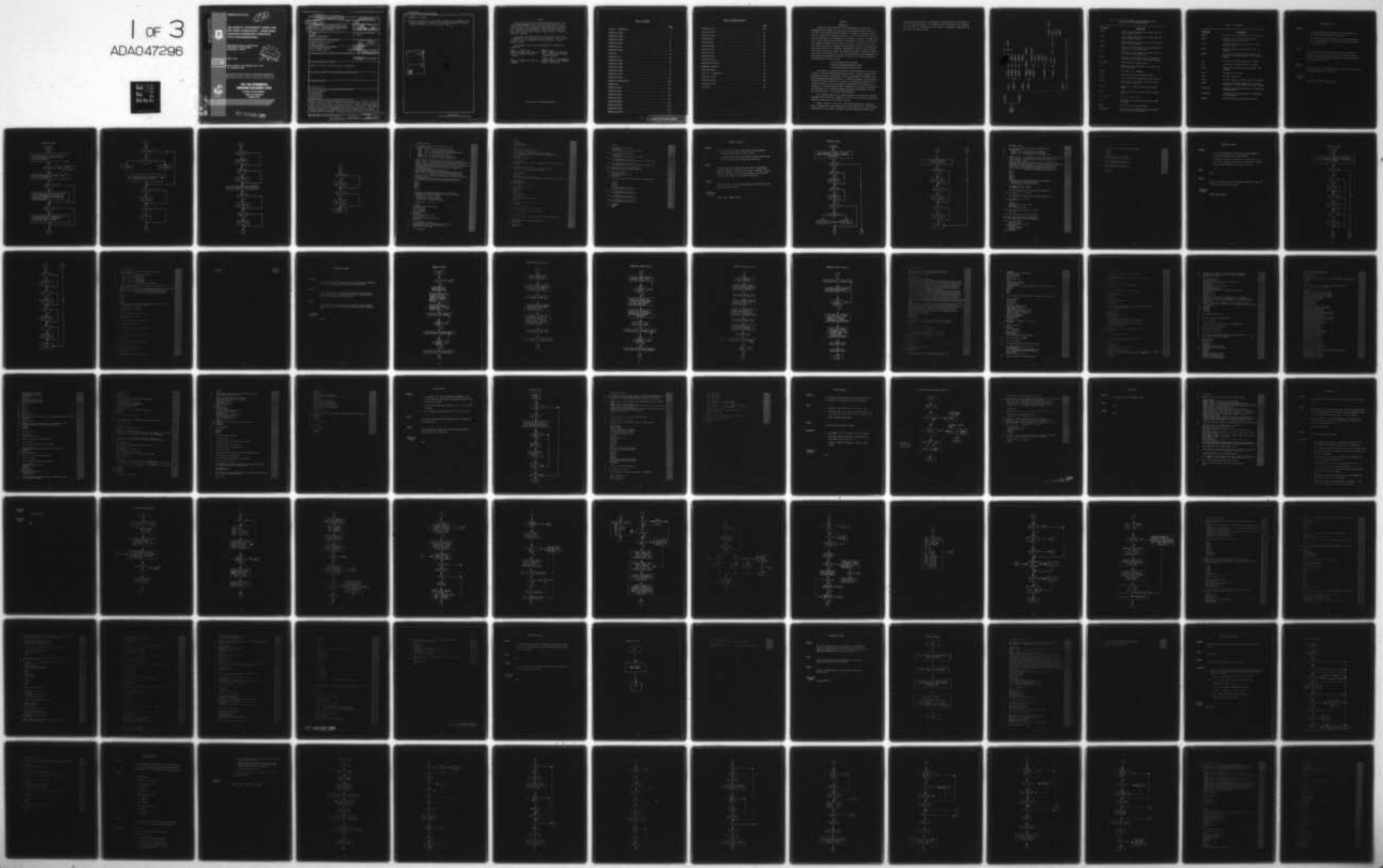
AD-A047 296 ARGONNE NATIONAL LAB ILL F/G 13/2
AIR QUALITY ASSESSMENT FOR AIR FORCE OPERATIONS - LONG-TERM EMI--ETC(U)
APR 77 D J BINGAMAN

UNCLASSIFIED CEEDO-TR-76-35 NL

1 OF 3
ADA047296



CEEDO-TR-76-35

AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - LONG-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION

ARGONNE NATIONAL LABORATORY
9700 SOUTH CASS AVENUE
ARGONNE IL 60439

APRIL 1977

CEEDO

FINAL REPORT FOR PERIOD JULY 1975 TO JANUARY 1977

Approved for public release; distribution unlimited

CIVIL AND ENVIRONMENTAL
ENGINEERING DEVELOPMENT OFFICE
(AIR FORCE SYSTEMS COMMAND)
TYNDALL AIR FORCE BASE
FLORIDA 32403

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (*When Data Entered*)

19 **REPORT DOCUMENTATION PAGE**	READ INSTRUCTIONS BEFORE COMPLETING FORM
18 1. ~~REPORT NUMBER~~ CEEDO-TR-76-35 · 2. GOVT ACCESSION NO.	3. ~~RECIPIENT'S CATALOG NUMBER~~ 9
4. ~~TITLE (and Subtitle)~~ 6 AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - LONG-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION.	5. ~~TYPE OF REPORT & PERIOD~~ COVERED Final Rep[redacted]t, 1 Jul[redacted] [redacted]75 [redacted]-1 Jan [redacted]77, 6. ~~PERFORMING ORG. REPORT NUMBER~~
7. AUTHOR(s) 10 Dorothy J./Bingaman	8. CONTRACT OR GRANT NUMBER(s) PO 76-003
9. PERFORMING ORGANIZATION NAME AND ADDRESS Argonne National Laboratory ✓ 9700 South Cass Avenue Argonne IL 60439	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 16 17 5A 62601/1900/5A03
11. CONTROLLING OFFICE NAME AND ADDRESS Det 1 (CEEDO) HQ ADTC/EC Tyndall AFB FL 32403	12. REPORT DATE 11 Apr[redacted] [redacted]77 13. NUMBER OF PAGES ~~252~~ 12 251p.
14. MONITORING AGENCY NAME & ADDRESS(*if different from Controlling Office*)	15. SECURITY CLASS. (~~of this report~~) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (*of this Report*)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)

18. SUPPLEMENTARY NOTES

Available in DDC

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Aircraft Assessment
Airport Models
Air Pollution
Dispersion Model
Computer Code

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

The Air Force contracted with Argonne National Laboratory to develop a series of computer programs designed to assess the air quality impact of Air Force operations at the airbase level. This report serves as a computer code documentation manual for the long-term emission/dispersion model of that effort. Descriptions of the computer codes corresponding to both the original version called the "Research Model" and the modified version called the "Applications Model" of the Long-Term Model are included. The manual contains flow charts, code listings, — over
continued . . .

20. ABSTRACT - continued

and brief descriptions of each routine contained in the model. It is intended primarily for readers with a computer programming background who wish to examine or alter the computer codes.

PREFACE

This report documents work performed during the period 1 July 1975 through December 1976 by Argonne National Laboratory. The technical work for this effort was performed under the auspices of the Air Force Civil Engineering Center (AFSC) which on 8 April 1977, reorganized into Detachment 1 (CEEDO) HQ ADTC, Tyndall Air Force Base, Florida, 32403. Captain Dennis F. Naugle, CEEDO/ECA, managed the program.

This report has been reviewed by the Information Officer and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Dennis F. Naugle
DENNIS F. NAUGLE, Capt, USAF, BSC
Chief, Environmental Modeling
Branch

PETER S. DALEY, Maj, USAF, BSC
Chief, Environmental Assessment
Research Division

JOSEPH S. PIZZUTO, Col, USAF, BSC
Commander

Peter A. Crowley
PETER A. CROWLEY, Maj, USAF, BSC
Director of Environics

(The reverse of this page is blank)

TABLE OF CONTENTS

TABLE OF CONTENTS (Contd.)

SECTION I

INTRODUCTION

Argonne National Laboratory (ANL) has developed an "Air Quality Assessment Model" (AQAM) for airbase operations under contract to the U.S. Air Force Civil Engineering Center (AFCEC). The model is designed to simulate the emission of pollutants from sources on an airbase and the dispersion of these pollutants in the atmosphere and calculates pollutant concentrations over a grid of ground level receptors. The model is comprised of four physically separate computer codes, of which three must be operated by the user. The fourth code prepares a magnetic tape containing long-term stability-time-wind roses for use by the long-term climatological type air pollution model. This code is operated on request by the USAF Environmental Technical Applications Center in Washington, D.C. The resultant magnetic tapes containing the climatological information is shipped to the user. The other three codes, developed by ANL, consist of the

- Source Inventory Model (SRCINV)
- Short-Term Emission/Dispersion Model
- Long-Term Emission/Dispersion Model

This report constitutes the computer code documentation for the third of these - the Long-Term Emission/Dispersion Model. Separate computer code documentation manuals (References 1 and 2) are available for SRCINV and the Short-Term Emission/Dispersion Model. A companion document to these reports, an Operator's Guide (Reference 3) consists of a detailed discussion of the various functional parts of the computer programs and the input/output requirements. A second companion report (Reference 4) discusses the technical and theoretical basis underlying AQAM and presents and describes equations and algorithms used in the various AQAM sub-models.

The intended purpose of the present document is to provide a computer programmer with sufficient information so that he can study the code and make changes or modifications to it where required.

Table 1 contains a list of all routines contained in the Long-Term Model in alphabetical order together with a brief description. More detailed descriptions of each routine, together with flow charts and computer code listings with comments that are intended to link listings to flow charts, are given on subsequent pages. It is hoped that this information, when combined with that given in References 1-4, will enable a programmer to understand and modify the code when desired.

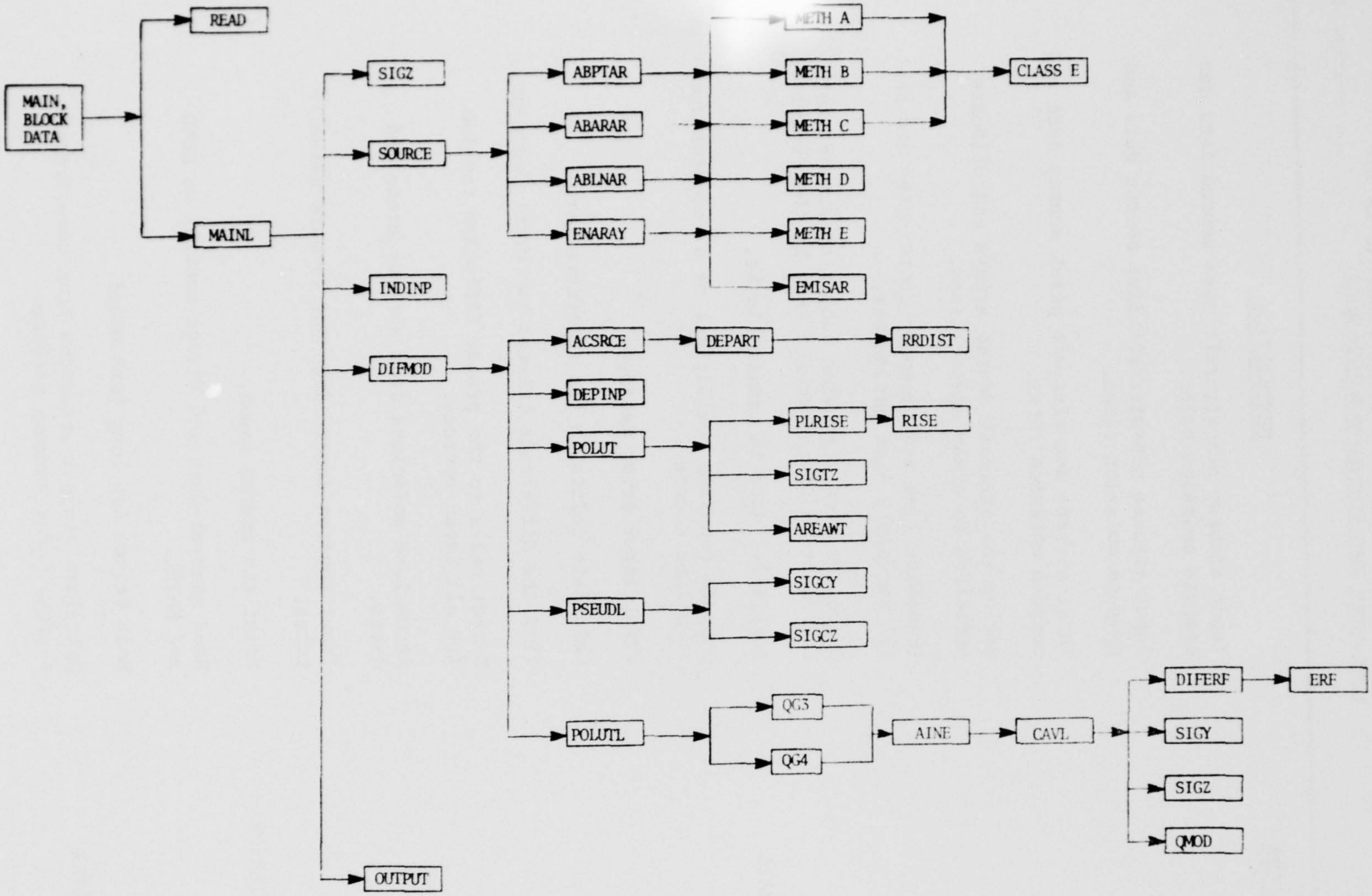


Figure 1. Schematic Flow Diagram of Long Term Research Model

TABLE 1. LIST OF ALL PROGRAMS AND SUB-PROGRAMS IN THE LONG-TERM EMISSION/DISPERSION MODEL

SUBROUTINE	DESCRIPTION
ABARAR	Input airbase non-aircraft area source data and compute emission rates.
ABLNAR	Input airbase non-aircraft line source data and compute emission rates.
ABPTAR	Input airbase non-aircraft point source data and compute emission rates.
ACSRCE	Set up the aircraft source arrays and allocate emissions to areas and/or lines.
AINE	Translate line and receptor coordinates and set all necessary line parameters.
AREAWT	Calculate area weighting factor to account for the fraction of the source seen by the receptor.
BLOCK DATA	Initialize data in common blocks.
CAVL	Compute coupling coefficient at a receptor due to a line source.
CLASSE	Print input error message.
DEPART	Calculate points in the departure path.
DIFERF	Find the difference between two error functions.
DIFMOD	Direct calls to the proper diffusion routine for all input sources.
EMISAR	Accumulate emissions from airbase areas and lines.
ENARAY	Input environ source data and compute emission rates.
INDINP/DEPINP	Print the source input.
MAIN	Read general data and direct control to READ and MAINL.
MAINL	Main driver for long term model.
METHA-METHE	Calculate diurnal emissions from non-aircraft sources using varying methods.

TABLE 1. (Concluded)

SUBROUTINE	DESCRIPTION
OUTPUT	Print pollutant concentrations at all receptors.
PLRISE	Calculate effective height and dispersion coefficients for a stack plume.
POLUT	Determine pollutant concentrations from point and area sources.
POLUTL	Determine pollutant concentrations from line sources.
PSEUDL	Call functions to find virtual distance from source to pseudo upwind point for all stability classes.
QG3	A three-point Gaussian quadrature procedure.
QG4	A four-point Gaussian quadrature procedure.
QMOD	Compute linear distribution of pollution along a runway.
READ	Read master source tape.
RISE	Calculate plume rise.
RRDIST	Calculate length of runway necessary for takeoff.
SIGTZ	Calculate the vertical dispersion or critical distance for all wind speed and stability classes.
SIGY/SIGCY	Calculate horizontal dispersion or corresponding virtual distance.
SIGZ/SIGCZ	Calculate vertical dispersion or corresponding virtual distance.
SOURCE	Driver for non-aircraft emission routines.

SUBROUTINE ABARAR

Purpose:

1. To read from the master source type all data needed to define airbase non-aircraft area sources.
2. To compute the emission rates due to evaporative hydrocarbons, space heating, off-road vehicles, and military and civilian vehicles.

Input:

If the diurnal distribution cards are input, an additional parameter, IOPT, is read here to choose the method of distribution of those evaporative hydrocarbons not using the default of a uniform distribution.

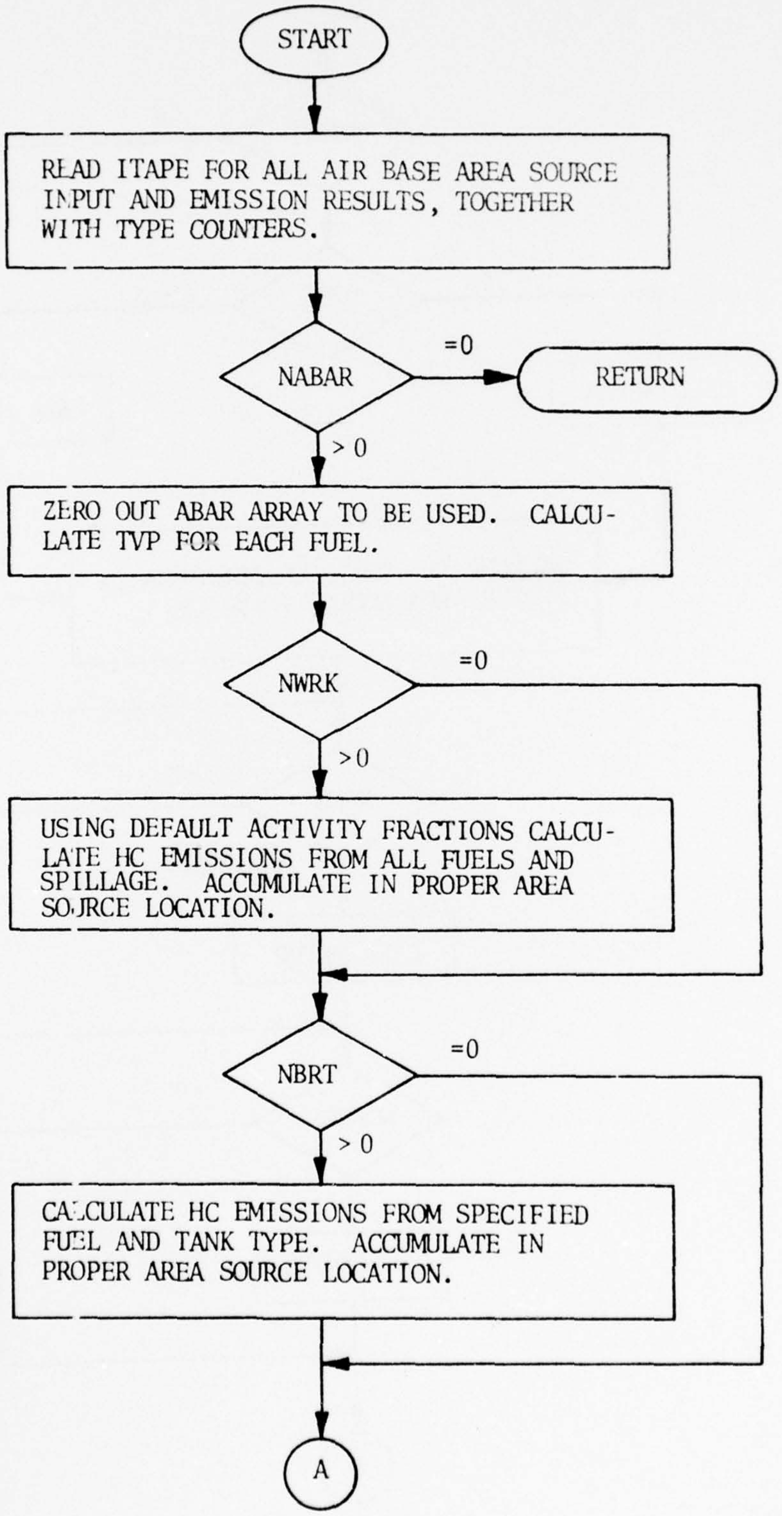
Output:

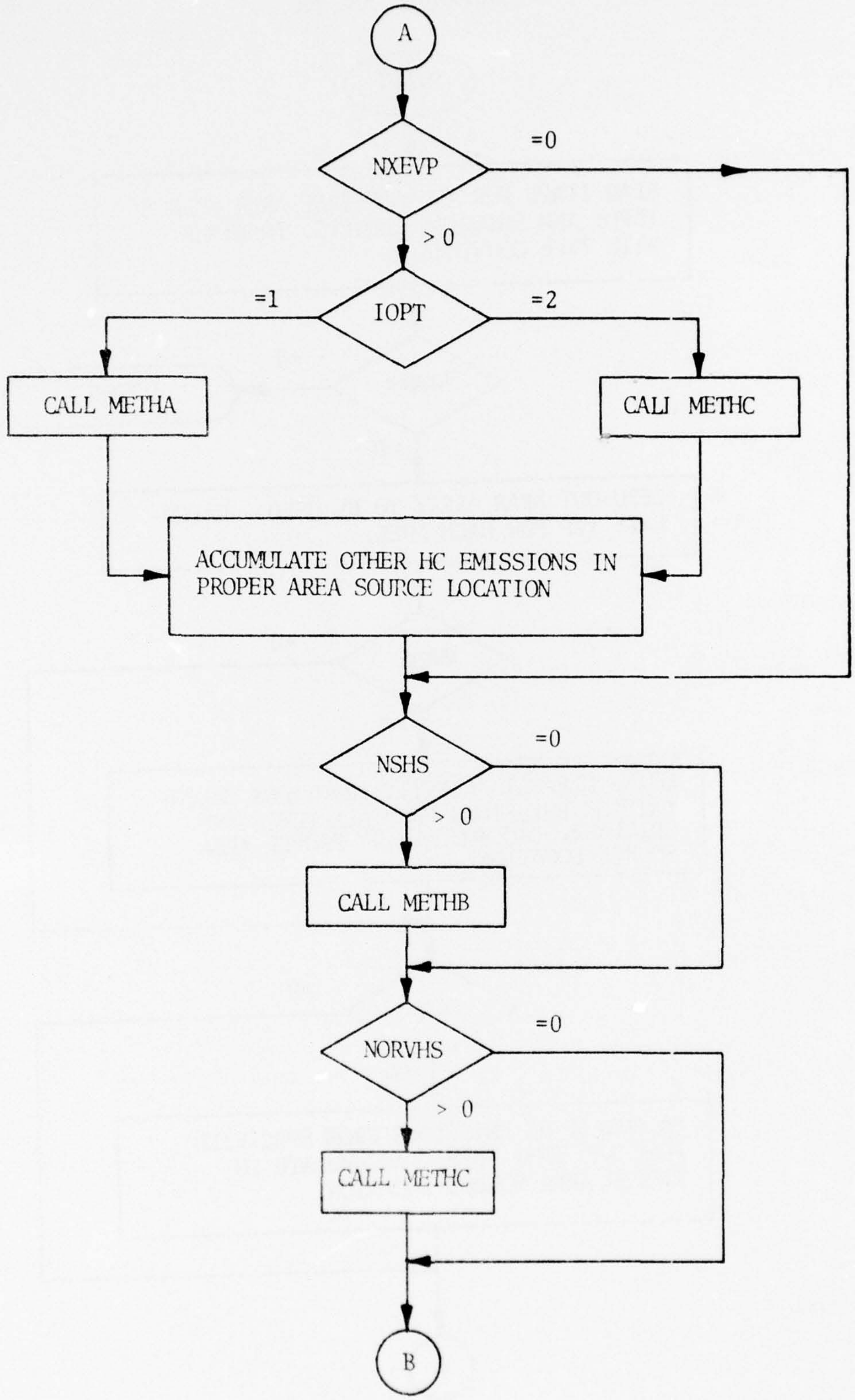
The array, ABAR, is filled with geometry and emission data for airbase non-aircraft area sources.

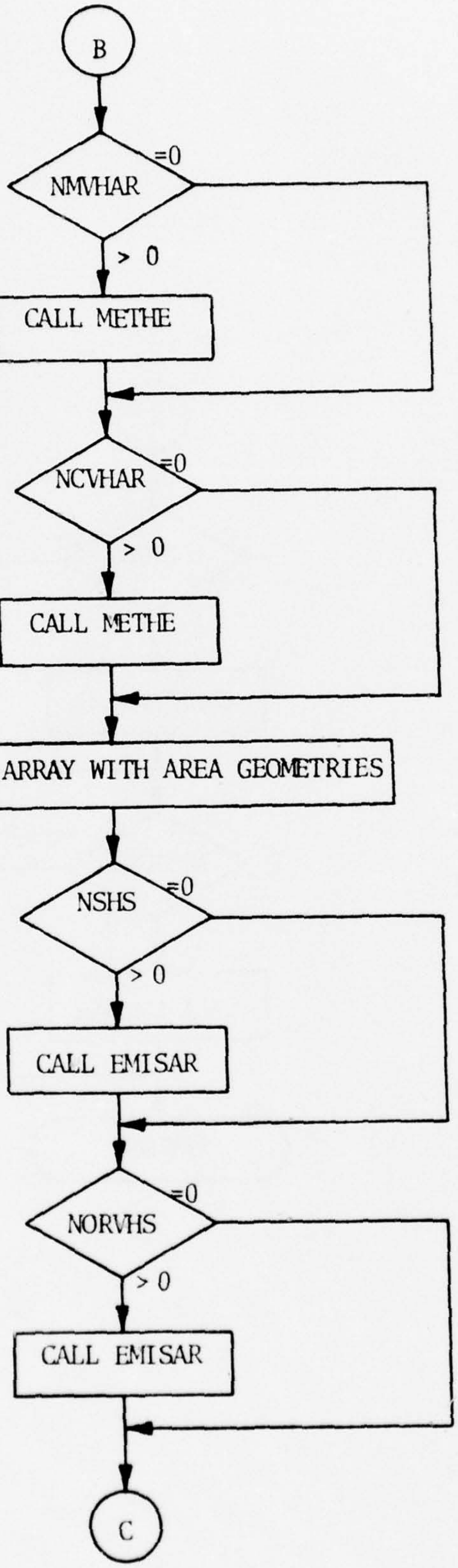
Subroutines Called:

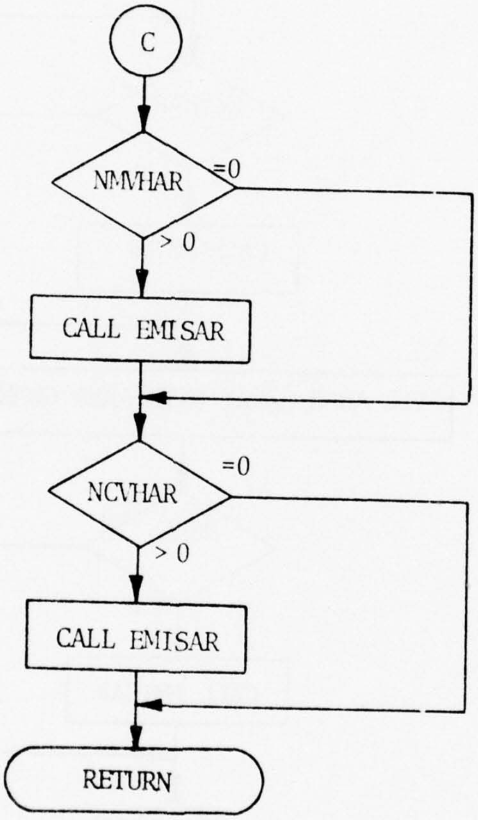
METHA, METHB, METHC, METHE, EMISAR

SUBROUTINE ABARAR


START
READ ITAPE FOR ALL AIR BASE AREA SOURCE INPUT AND EMISSION RESULTS, TOGETHER WITH TYPE COUNTERS.
NABAR
=0
RETURN
> 0
ZERO OUT ABAR ARRAY TO BE USED. CALCULATE TVP FOR EACH FUEL.
NWRK
=0
>0
USING DEFAULT ACTIVITY FRACTIONS CALCULATE HC EMISSIONS FROM ALL FUELS AND SPILLAGE. ACCUMULATE IN PROPER AREA SOURCE LOCATION.
NBRT
=0
> 0
CALCULATE HC EMISSIONS FROM SPECIFIED FUEL AND TANK TYPE. ACCUMULATE IN PROPER AREA SOURCE LOCATION.
A


A
NXEVP
=0
> 0
IOPT
=1
=2
CALL METHA
CALI METHC
ACCUMULATE OTHER HC EMISSIONS IN PROPER AREA SOURCE LOCATION
NSHS
=0
> 0
CALL METHB
NORVHS
=0
> 0
CALL METHC
B


B
NMVHAR
=0
> 0
CALL METHE
NCVHAR
=0
> 0
CALL METHE
FILL ABAR ARRAY WITH AREA GEOMETRIES
NSHS
=0
> 0
CALL EMISAR
NORVHS
=0
> 0
CALL EMISAR
C


C
NMVHAR
=0
> 0
CALL EMISAR
NCVHAR
=0
> 0
CALL EMISAR
RETURN

```
      SUBFOUTINE ABAFAF
C
C     TIHS ROUTINE CCMPUTES THE EMISSION RATES FOR
C     AIL AIRBASE AREAS
C        NWRK   = NO. OF HYDROCARBON WORKING LOSSES
C        NBFT   = NC. OF HYDROCARBON BREATHING LOSSES
C        NXEVP  = NC. OF CTHER EVAPORATIVE HYDROCARBON SOURCES
C        NSHS   = NO. OF SPACE HEATING SOURCES
C        NCRVHS = NO. CF OFF-ROAD VEHICLE SOURCES
C        NMVHAR = NO. OF MILITARY VEHICLE AREA SOURCES
C        NCVHAR = NO. OF CIVILIAN VEHICLE AREA SOURCES
C
      CCMMCN /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      CCMMCN / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMCN /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHP(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVEFHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      CCMMCN/JUNK/DAYS,LSRCE,NSPCE,SORCE(17,300),SORGM(10,200)
     . ,LCC1,LOC2,NGEOM,IPT
      COMMON/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR
      CCMMCN /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                    ABPT(16,150),ABAR(11,100),ABLN(14,100)
      DIMENSION ABARGM(7,100),HCWRK(10,50),HCBRT(5,100),HCEVP(3,50),
     . FLHCUR(7),TVP(7)
      EQUIVALENCE (SORGM(1),ABARGM(1)),(SORGM( 701),HCWRK(1)),
     .      (SORGM(1201),HCBRT(1)), (SORGM(1701),HCEVP(1))
      LCC1=2
      LCC2=2
      NGECM=0
      IPT=0
      NSRCE=0
      I1=17
      I2=300
C
      READ(ITAPE) NABAR,NTOT,NWRK,NBRT,NXEVP,NSHS,NORVHS,
     .  NMVHAR,NCVHAR,NABARS,((ABARGM(I,N),I=1,7),N=1,NABAR),
     . ((HCWRK(I,N),I=1,10),N=1,NWRK),
     .  ((HCBRT(I,N),I=1,5),N=1,NBFT),
     .  ((HCEVP(I,N),I=1,3),N=1,NXEVP),
     .  ((SCRCE(I,N),I=1,NTCT),N=1,NABARS)
C
      IF (NABAR.EQ.0) GO TO 1100
C
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1+1
      DC 1C N=1,NABAR
      DC 10 I=1,NPLTS
      ABAR(I+5,N)=0.0
   10 CCNTINUE
      T=5./9.*(TMBAR-32.0)+273.
      DC 20 J=1,7
      TVP(J)=EXP(ALPHA(J)-BETA(J)/T)
   20 CONTINUE
C
      IF (NWRK.EQ.0) GO TO 100
C     USING DEFAULT ACTIVITY FRACTIONS CALCULATE HC
C     EMISSIONS FROM ALL FUELS AND SPILLAGE.
C     ACCUMULATE IN ABAR ARRAY
C
      DC 50 N=1,NWRK
      HC=0.
      FFC=C.
      DC 40 J=1,7
      FLHOUR(J)=0.
      DC 30 I=IHR1,NHI
      II=I
      IF(I.GT.24) II=I-24
   30 FLHOUR(J)=FLHOUR(J)+FLHR(II,J)
      FLHOUR(J)=FLHOUR(J)/HRS
      FFC=FRC+FLHOUR(J)*FLMC(IMONTH,J)*FLDY(IDAY,J)
      HC=HC+HCWRK(J+2,N)*TVP(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
     .  *FLHOUR(J)*7./DAYS
   40 CONTINUE
      FRC=FFC/4.*7./DAYS
C
      J=HCWRK(2,N)
      ABAR(7,J)=APAR(7,J) + (HC + HCWRK(10,N) * FRC) *(1.E+6/3.6)
   50 CCNTINUE
C
  100 IF (NERT.EQ.0) GO TO 200
C     CALCULATE HC EMISSIONS FROM SPECIFIED FUEL AND
C     TANK TYPES.  ACCUMULAIE IN ABAR ARRAY
C
      DC 110 N=1,NBRT
      J=HCBRT(3,N)
      EX=0.68
      IF (HCBRT(4,N).EQ.2.) EX=0.70
      HC=HCBRT(5,N)*(TVP(J)/(14.7-TVP(J)))**EX*(1.E+6/(3.6*24.*365.))
C
      J=HCBRT(2,N)
      ABAR(7,J)=ABAR(7,J)+HC
  110 CCNTINUE
C
  200 IF (NXEVP.EQ.0) GO TO 300
      ICLASS=110
      NTEMP=NELTS
      NELTS=1
      LCC1=3
      NSRCE=NABARS
      DC 210 N=1,NXEVP
      DC 210 I=1,3
      SCRCE(I,NABARS+N)=HCEVP(I,N)
  210 CCNTINUE
      ICPT=1
      IF (JFLAG.EQ.0) READ 2,IOPT
    2 FCRMAT(I4)
      GC TC (220,230),IOPT
C
  220 CALL METHA(NXEVP,SORCE,I1,I2,ICLASS)
      GC TO 240
  230 CALL METHC(NXEVP,SORCE,I1,I2,ICLASS)
C
  240 DC 250 N=1,NXEVP
C
C     ACCUMULATE CTHER EVAPCRATIVE HC EMISSIONS IN ABAR ARRAY
C
      J=HCEVP(2,N)
      ABAR(7,J)=ABAR(7,J)+SCRCE(3,NABARS+N)
  250 CONTINUE
C
      NELTS=NTEMP
      NSRCE=0
      LOC1=2
C
  300 IF (NSHS.EQ.0) GO TO 400
      ICLASS=111
      CALL METHB(NSHS,SORCE,I1,I2,ICLASS)
C
  400 IF (NORVHS.EQ.0) GO TO 500
      ICLASS=112
      CALL METHC(NORVHS,SORCE,I1,I2,ICLASS)
C
  500 IF (NMVHAR.EQ.0) GO TO 600
      CALL METHE(NMVHAR,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)
C
  600 IF (NCVHAR.EQ.0) GO TO 700
      CALL METHE(NCVHAR,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)
C
C****EMISSIONS ARE NOW IN MICROGRAMS/SEC
C     FILL ABAR ARRAY WITH AREA GEOMETRIES
C
  700 DO 710 N=1,NABAR
      DO 710 I=1,5
      ABAR(I,N)=ABARGM(I+2,N)
  710 CONTINUE
C
C     FILL ABAR ARRAY WITH THE NON-EVAP HC EMISSION DATA
C
      I1=11
      I2=100
      NSRCE=0
      LOC1=5
      IF (NSHS.EQ.0) GO TO 800
      CALL EMISAR(NSHS,ABAR,I1,I2)
C
  800 IF (NORVHS.EQ.0) GO TO 900
      CALL EMISAR(NORVHS,ABAR,I1,I2)
C
  900 IF (NMVHAR.EQ.0) GO TO 1000
      CALL EMISAR(NMVHAR,ABAR,I1,I2)
C
 1000 IF (NCVHAR.EQ.0) GO TO 1100
      CALL EMISAR(NCVHAR,ABAR,I1,I2)
C
 1100 CONTINUE
      RETURN
      END
```

SUBROUTINE ABLNAR

Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft line sources.
2. To compute the emission rates due to military and civilian vehicle line and other airbase line activities.

Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those other airbase line activities not using the default of a uniform distribution.

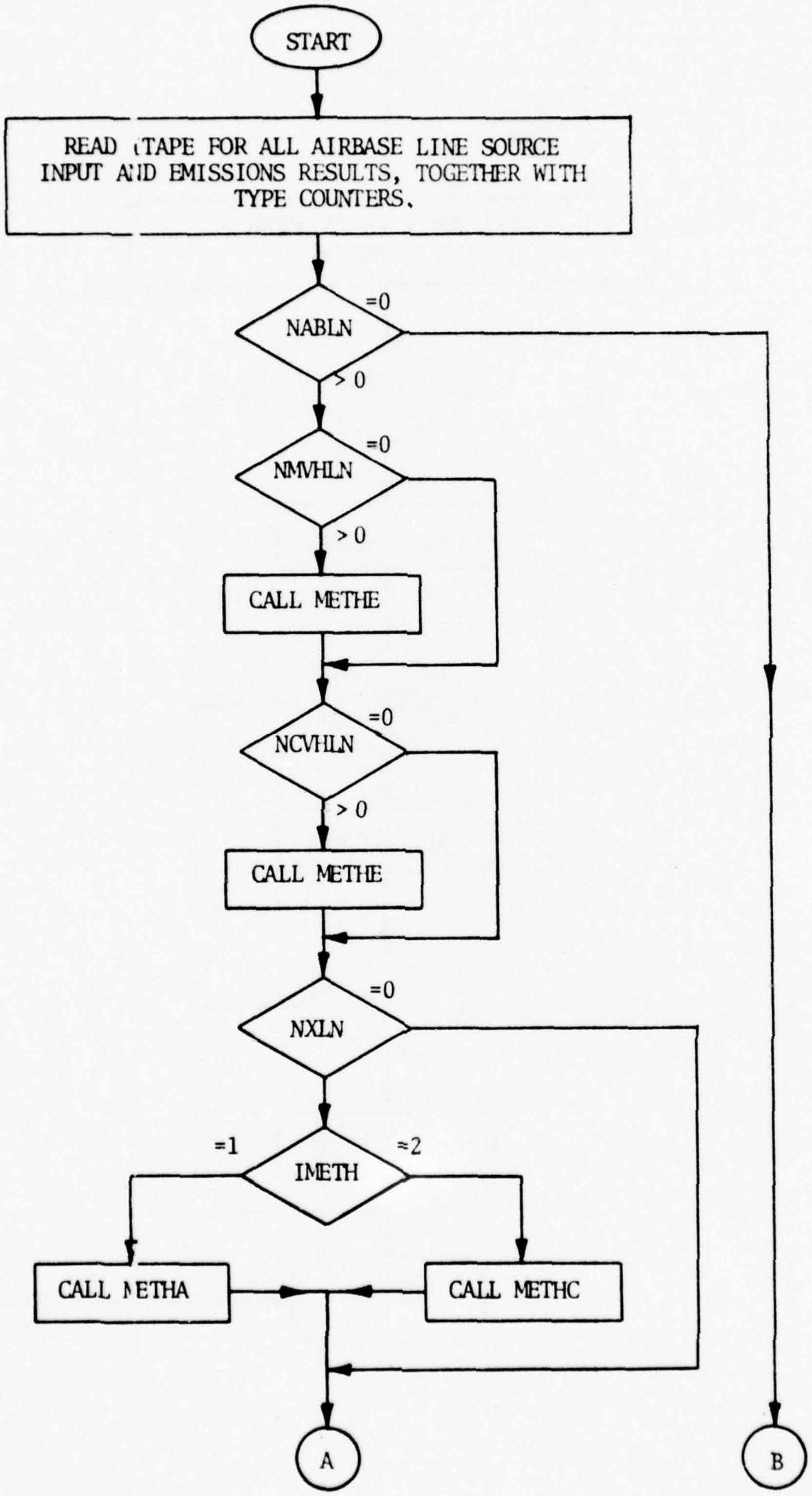
Output:

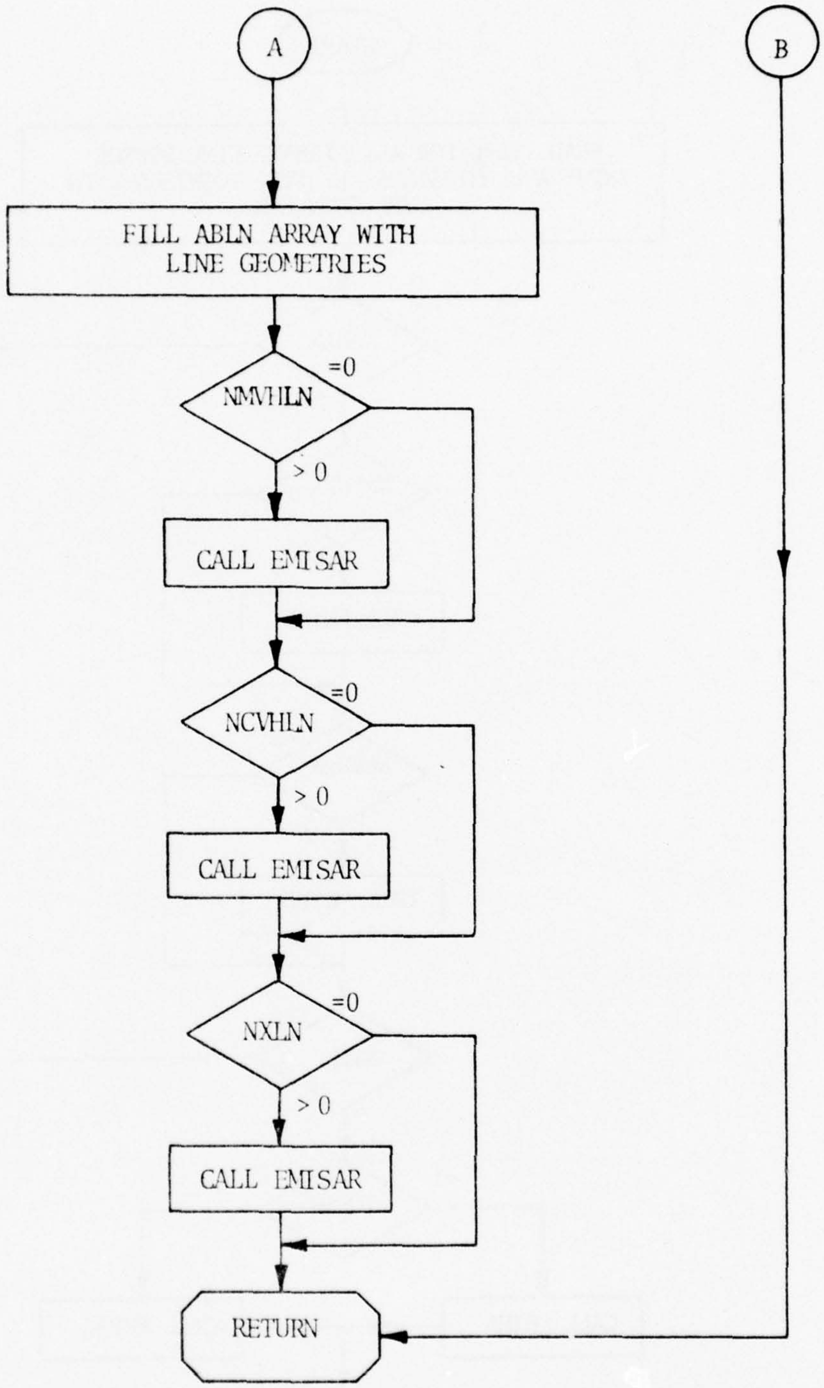
The array, ABLN, is filled with geometry and emission data for non-aircraft line sources.

Subroutines Called:

METHA, METHC, METHE, EMISAR

SUBROUTINE ABLNAR


START
READ TAPE FOR ALL AIRBASE LINE SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS.
NABLN
=0
> 0
NMVHLN
=0
> 0
CALL METHE
NCVHLN
=0
> 0
CALL METHE
NXLN
=0
IMETH
=1
=2
CALL METHA
CALL METHC
A
B


A
B
FILL ABLN ARRAY WITH LINE GEOMETRIES
NMVHLN
=0
> 0
CALL EMISAR
NCVHLN
=0
> 0
CALL EMISAR
NXLN
=0
> 0
CALL EMISAR
RETURN

```
      SUBROUTINE ABLNAR
C
C     THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL
C     AIRBASE LINES
C        NMVHLN = NO. OF MILITARY LINE ACTIVITIES
C        NCVHLN = NO. OF CIVILIAN LINE ACTIVITIES
C        NXLN   = NO. OF OTHER AIR BASE LINE ACTIVITIES
C
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                     ABPT(16,150),ABAR(11,100),ABLN(14,100)
      LOC1=2
      LOC2=2
      I1=17
      I2=300
      NGEOM=0
      IPT=0
      NSRCE=0
      READ(ITAPE) NABLN,NTOT,NMVHLN,NCVHLN,NXLN,NABLNS,
     .  ((SORGM(I,N),I=1,10),N=1,NABLN),
     .  ((SORCE(I,N),I=1,NTOT),N=1,NABLNS)
C
      IF (NABLN.EQ.0) GO TO 600
      IF (NMVHLN.EQ.0) GO TO 100
C
      CALL METHE(NMVHLN,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)
C
  100 IF (NCVHLN.EQ.0) GO TO 200
C
      CALL METHE(NCVHLN,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)
C
  200 IF (NXLN.EQ.0) GO TO 300
      ICLASS=117
C
      IMETH=1
      IF (JFLAG.EQ.0) READ 1,IMETH
    1 FORMAT(I4)
      GO TO (210,220),IMETH
C
  210 CALL METHA(NXLN,SORCE,I1,I2,ICLASS)
      GO TO 300
C
  220 CALL METHC(NXLN,SORCE,I1,I2,ICLASS)
C
C****EMISSIONS ARE NOW IN MICROGRAMS/SEC
C     FILL ABLN ARRAY WITH LINE GEOMETRIES
C
  300 DO 320 N=1,NABLN
      DO 310 I=1,8
      ABLN(I,N)=SORGM (I+2,N)
  310 CONTINUE
      DO 320 I=1,NPLTS
      ABLN(I+8,N)=0.0
  320 CONTINUE
C
```

```
C     FILL ABLN ARRAY WITH LINE EMISSION DATA
C
      NSRCE=0
      LOC1=8
      I1=14
      I2=100
      IF (NMVHLN.EQ.0) GO TC 400
      CALL EMISAR(NMVHLN,ABLN,I1,I2)
C
  400 IF (NCVHLN.EQ.0) GO TO 500
      CALL EMISAR(NCVHLN,ABLN,I1,I2)
C
  500 IF (NXLN.EQ.0) GO TO 600
      CALL EMISAR(NXLN,ABLN,I1,I2)
C
  600 CONTINUE
      RETURN
      END
```

SUBROUTINE ABPTAR

Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft point sources.
2. To compute the emission rates due to training fires, test cells, run-up stands, power plants, incinerators, storage tanks and other airbase point source activities.

Input:

None

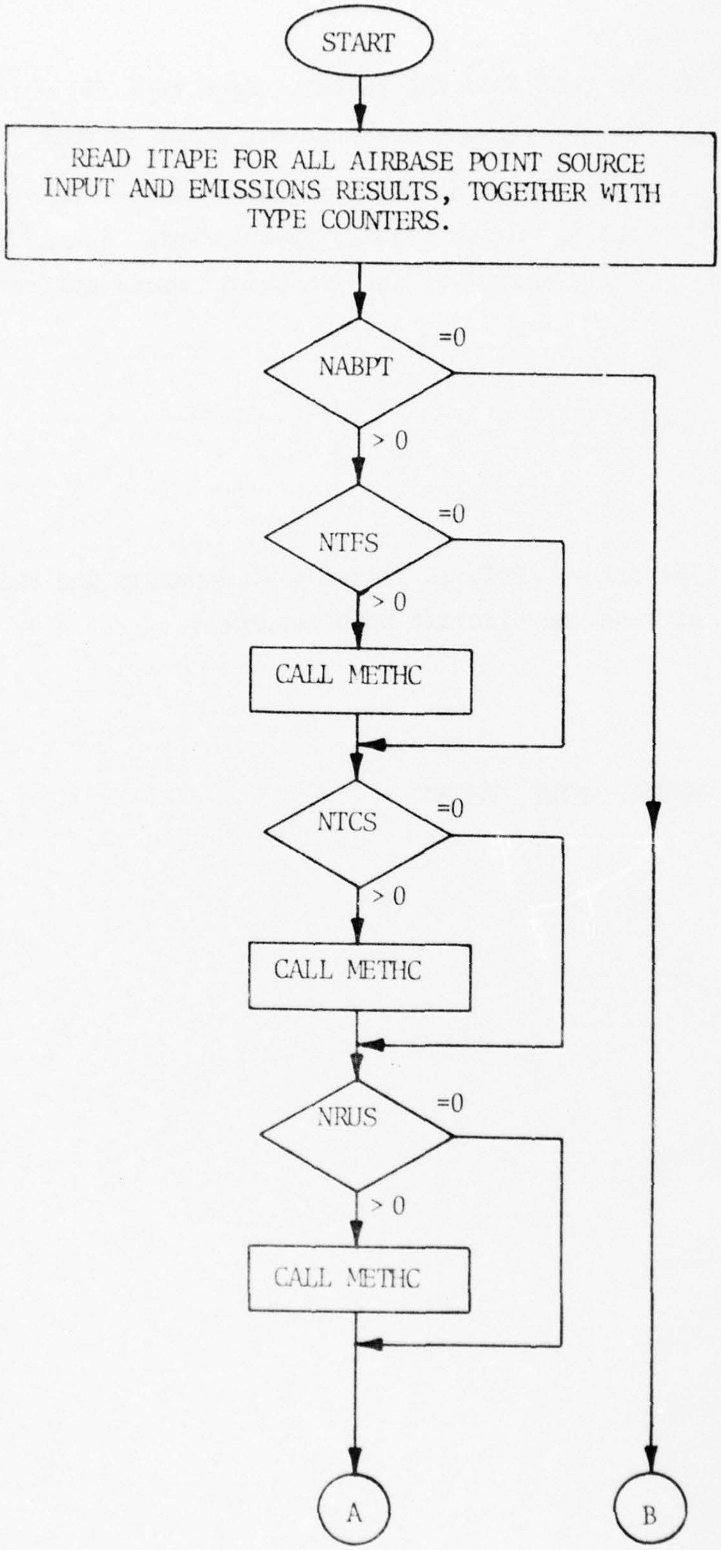
Output:

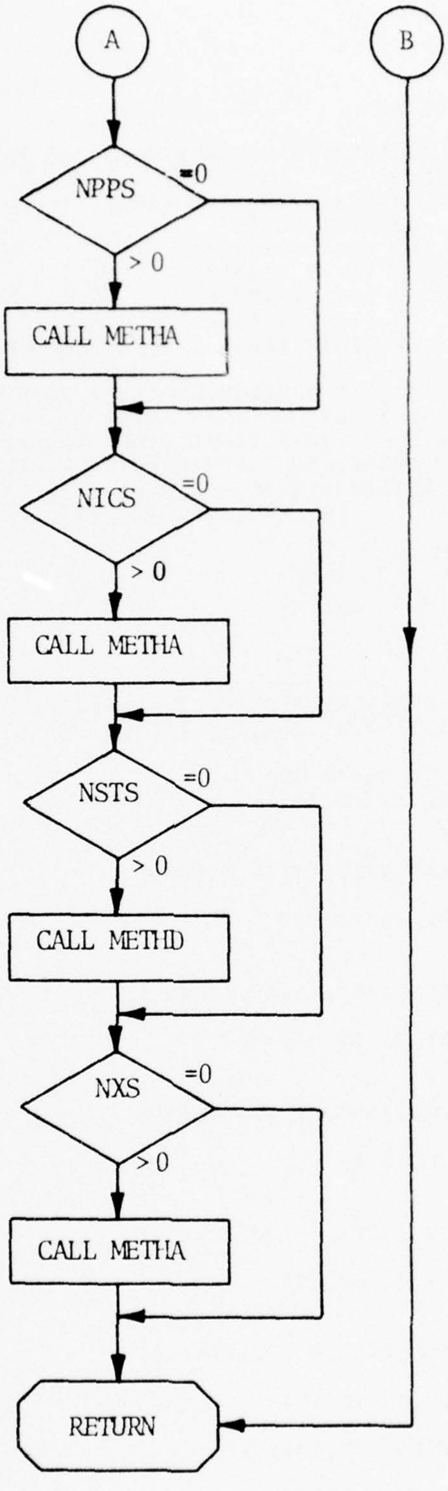
The array, APBT, is filled with geometry and emission data for airbase non-aircraft point sources.

Subroutines Called:

METHA, METHC, METHD

SUBROUTINE ABPTAR


START
READ ITAPE FOR ALL AIRBASE POINT SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS.
NABPT
=0
> 0
NTFS
=0
> 0
CALL METHC
NTCS
=0
> 0
CALL METHC
NRUS
=0
> 0
CALL METHC
A
B


A
B
NPPS
=0
> 0
CALL METHA
NICS
=0
> 0
CALL METHA
NSTS
=0
> 0
CALL METHD
NXS
=0
> 0
CALL METHA
RETURN

```
      SUBROUTINE ABPTAR
C
C     THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL
C     AIRBASE POINTS
C        NTFS = NO. OF TRAINING FIRE SITES
C        NTCS = NO. OF TEST CELLS
C        NRUS = NO. OF RUN-UP STANDS
C        NPPS = NO. OF POWER PLANTS
C        NICS = NO. OF INCINERATORS
C        NSTS = NO. OF STORAGE TANKS
C        NXS  = NO. OF OTHER AIRBASE POINT SOURCES
C
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10 200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                    ABPT(16,150),ABAR(11,100),ABLN(14,100)
      LOC1=10
      LOC2=11
      NGEOM=9
      IPT=1
      NSRCE=0
      I1=16
      I2=200
      READ(ITAPE) NABPT,NTOT,NTFS,NTCS,NRUS,NPPS,NICS,NSTS,NXS,
     .  ((SORCE(I,N),I=1,NTOT),N=1,NABPT)
C
      IF (NABPT.EQ.0) GO TO 700
      IF (NTFS.EQ.0) GO TO 100
      ICLASS=101
C
      CALL METHC(NTFS,ABPT,I1,I2,ICLASS)
C
  100 IF (NTCS.EQ.0) GO TO 200
      ICLASS=102
C
      CALL METHC(NTCS,ABPT,I1,I2,ICLASS)
C
  200 IF (NRUS.EQ.0) GO TO 300
      ICLASS=103
C
      CALL METHC(NRUS,ABPT,I1,I2,ICLASS)
C
  300 IF (NPPS.EQ.0) GO TO 400
      ICLASS=104
C
      CALL METHA(NPPS,ABPT,I1,I2,ICLASS)
C
  400 IF (NICS.EQ.0) GO TO 500
      ICLASS=105
C
      CALL METHA(NICS,ABPT,I1,I2,ICLASS)
C
  500 IF (NSTS.EQ.0) GO TO 600
C
      CALL METHD(NSTS,ABPT,I1,I2)
C
  600 IF (NXS.EQ.0) GO TO 700
      ICLASS=107
C
      CALL METHA(NXS,ABPT,I1,I2,ICLASS)
```

```
C                                                                       ABPTR062
  700 RETURN                                                            ABPTR063
      END                                                               ABPTR064
```

SUBROUTINE ACSRCE

Purpose:

To set up the aircraft source arrays to be used by the dispersion routines for calculating ground level concentrations.

Input:

Basic aircraft data, airbase activity data, points in arrival-departure paths and in training flight paths, meteorological conditions, time period of calculation.

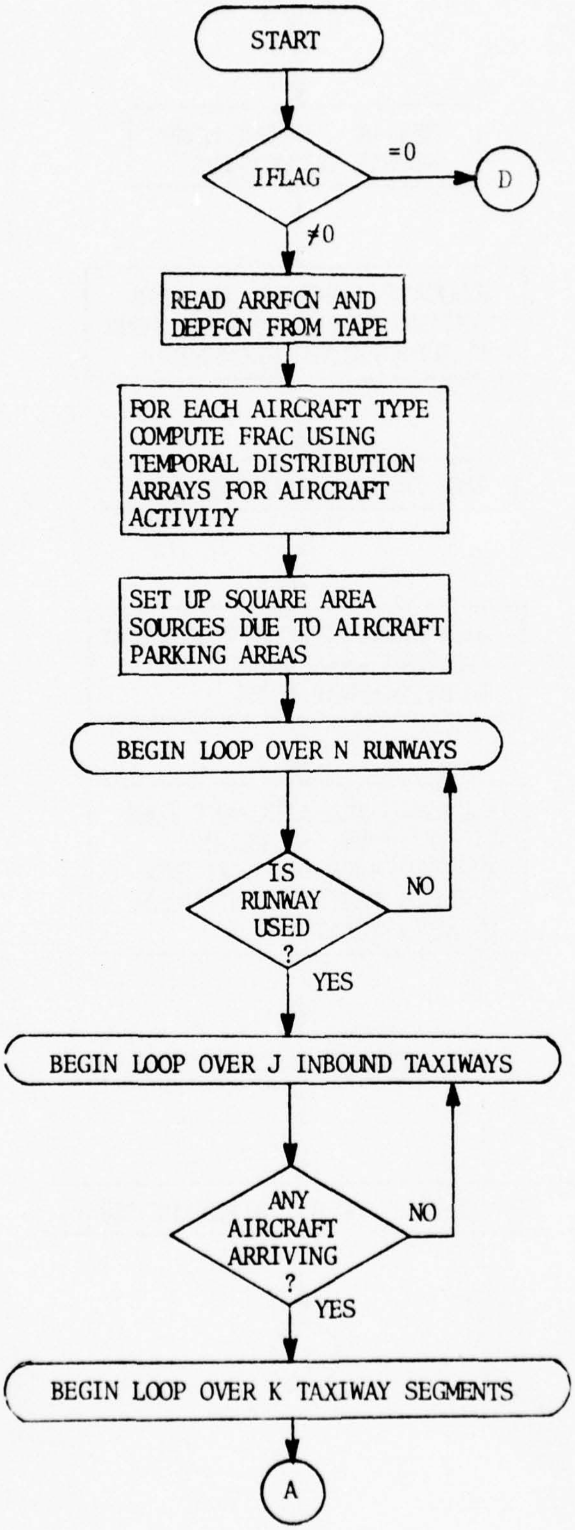
Output:

The arrays ACPT, ACLN and ACAR to contain all source information necessary to calculate dispersion and pollutant concentrations.

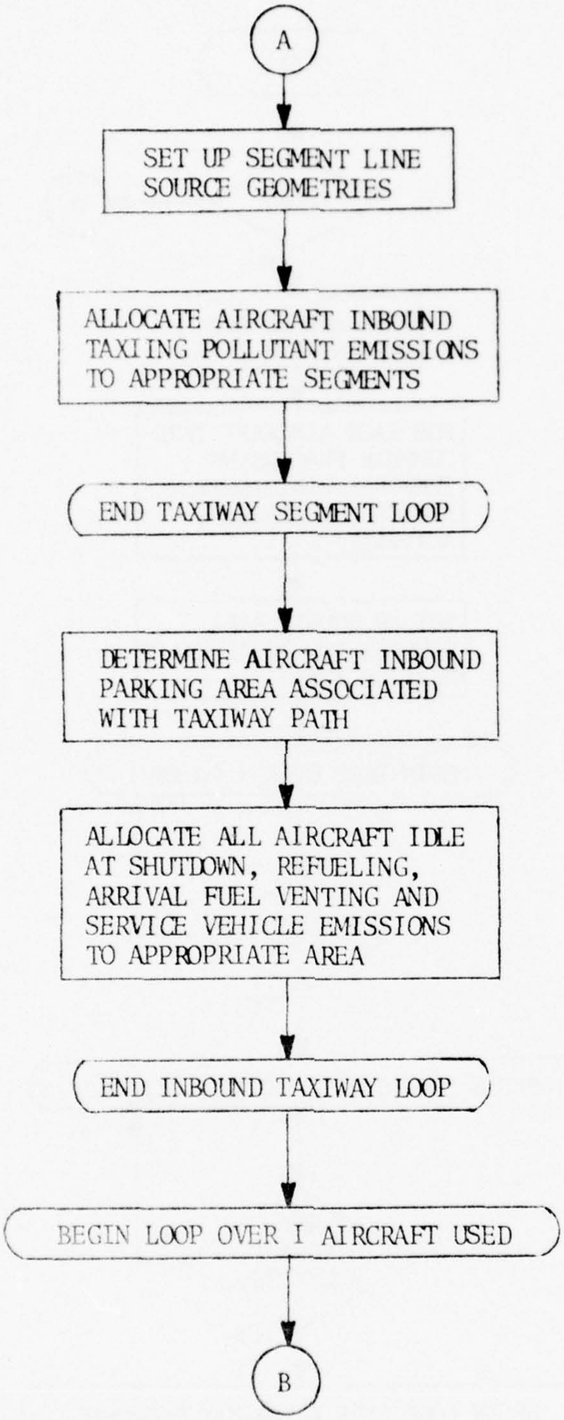
Subroutine Called:

DEPART

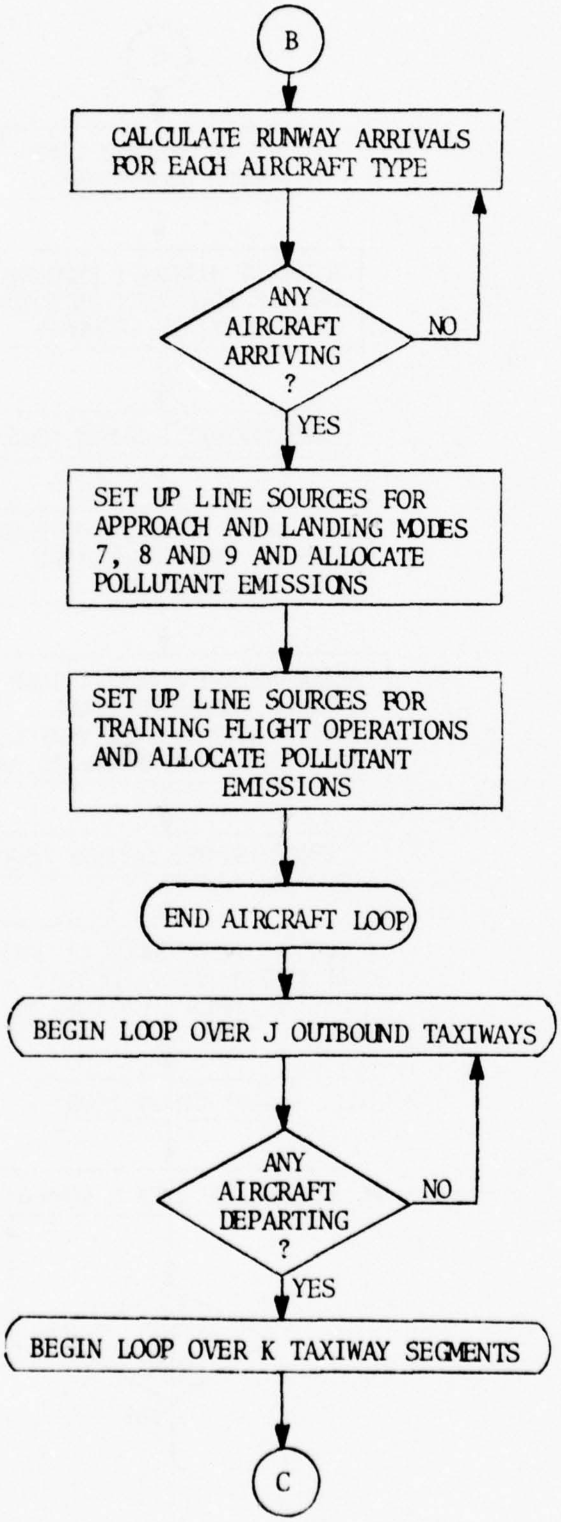
SUBROUTINE ACSRCE


START
IFLAG
=0
D
≠0
READ ARRFCN AND DEPFCN FROM TAPE
FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY
SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS
BEGIN LOOP OVER N RUNWAYS
IS RUNWAY USED ?
NO
YES
BEGIN LOOP OVER J INBOUND TAXIWAYS
ANY AIRCRAFT ARRIVING ?
NO
YES
BEGIN LOOP OVER K TAXIWAY SEGMENTS
A

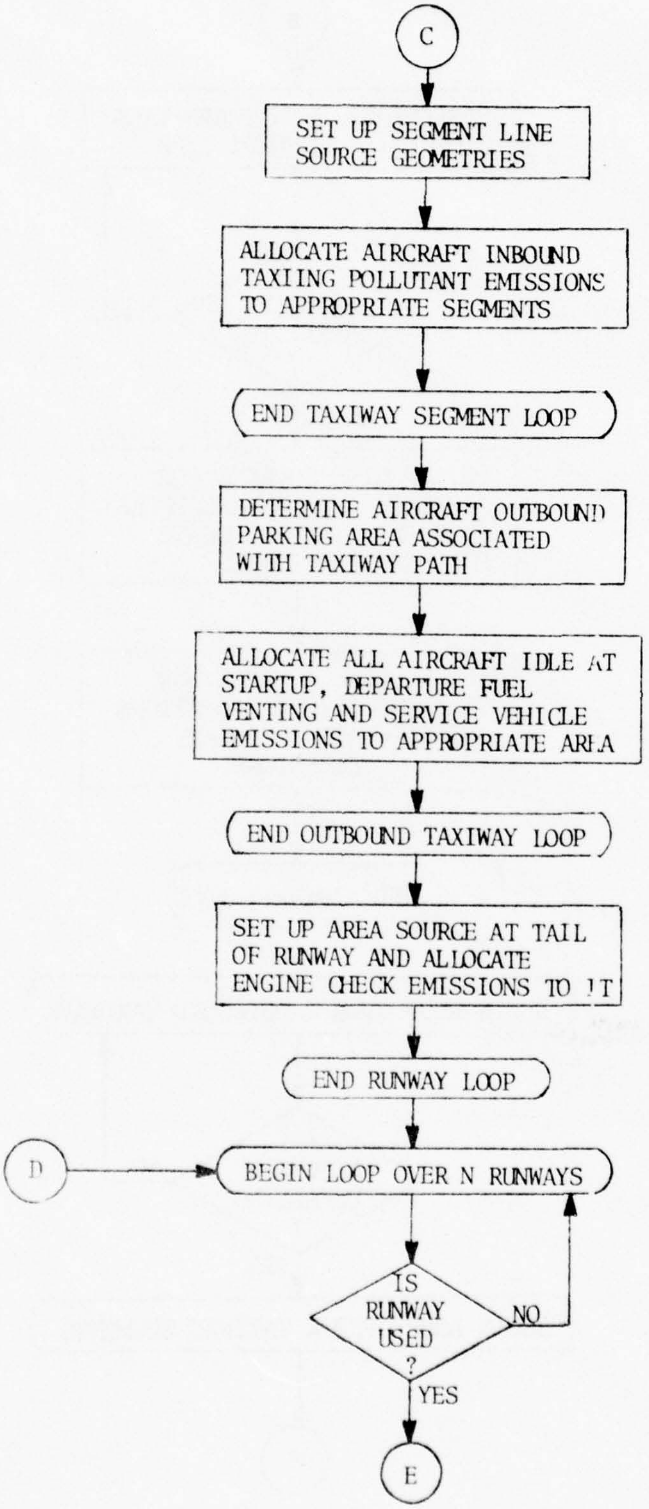
SUBROUTINE ACSRCE (Cont'd.)


A
SET UP SEGMENT LINE SOURCE GEOMETRIES
ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS TO APPROPRIATE SEGMENTS
END TAXIWAY SEGMENT LOOP
DETERMINE AIRCRAFT INBOUND PARKING AREA ASSOCIATED WITH TAXIWAY PATH
ALLOCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING, ARRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
END INBOUND TAXIWAY LOOP
BEGIN LOOP OVER I AIRCRAFT USED
B

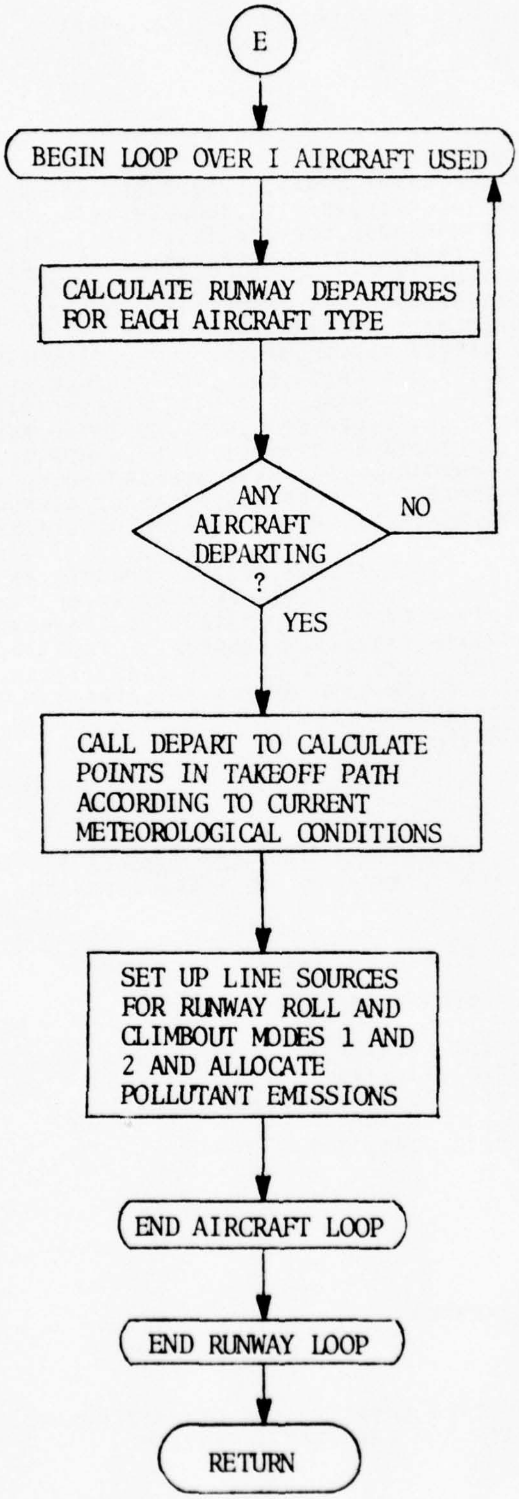
SUBROUTINE ACSRCE (Cont'd.)


B
CALCULATE RUNWAY ARRIVALS FOR EACH AIRCRAFT TYPE
ANY AIRCRAFT ARRIVING ?
NO
YES
SET UP LINE SOURCES FOR APPROACH AND LANDING MODES 7, 8 AND 9 AND ALLOCATE POLLUTANT EMISSIONS
SET UP LINE SOURCES FOR TRAINING FLIGHT OPERATIONS AND ALLOCATE POLLUTANT EMISSIONS
END AIRCRAFT LOOP
BEGIN LOOP OVER J OUTBOUND TAXIWAYS
ANY AIRCRAFT DEPARTING ?
NO
YES
BEGIN LOOP OVER K TAXIWAY SEGMENTS
C

SUBROUTINE ACSRCE (Cont'd.)


C
SET UP SEGMENT LINE SOURCE GEOMETRIES
ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS TO APPROPRIATE SEGMENTS
END TAXIWAY SEGMENT LOOP
DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED WITH TAXIWAY PATH
ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
END OUTBOUND TAXIWAY LOOP
SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE ENGINE CHECK EMISSIONS TO IT
END RUNWAY LOOP
D
BEGIN LOOP OVER N RUNWAYS
IS RUNWAY USED ?
NO
YES
E

SUBROUTINE ACSRCE (Cont'd.)


E
BEGIN LOOP OVER I AIRCRAFT USED
CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE
ANY AIRCRAFT DEPARTING ?
NO
YES
CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING TO CURRENT METEOROLOGICAL CONDITIONS
SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 AND ALLOCATE POLLUTANT EMISSIONS
END AIRCRAFT LOOP
END RUNWAY LOOP
RETURN

```
      SUBROUTINE ACSRCE                                                 ACSRC000
C                                                                       ACSRC001
C     THIS ROUTINE SETS UP THE AIRCRAFT SOURCE ARRAYS                   ACSRC002
C     AND ALLOCATES THE POLLUTANT EMISSIONS TO THE                      ACSRC003
C     APPROPRIATE AREA OR LINE                                          ACSRC004
C                                                                       ACSRC005
      REAL LNDSPD                                                       ACSRC006
      INTEGER  ENGNO                                                    ACSRC007
      COMMON /RECPT/ MRECPT,MAXFIL                                      ACSRC008
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,    ACSRC009
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),  ACSRC010
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)    ACSRC011
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),     ACSRC012
     . LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),       ACSRC013
     . COSPD2(8),SRTUPI(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),     ACSRC014
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)                        ACSRC015
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),   ACSRC016
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),    ACSRC017
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),   ACSRC018
     . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),      ACSRC019
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),   ACSRC020
     . IOBSEG(16,8,6),IOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),      ACSRC021
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)ACSRC022
      COMMON / MET / WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF, ACSRC023
     1 TEMK                                                             ACSRC024
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) ACSRC025
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMIMO(13),       ACSRC026
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), ACSRC027
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1         ACSRC028
      COMMON /PERIOD/ IMO   ,NODAYS,IDY ,IHR1,IHR2,IFLAG,JFLAG          ACSRC029
      DIMENSION IACAR(2,18),FRAC(8),PARFCT(18),APARSQ(6,3),NQ(25)       ACSRC030
      XP(XO,YC,W)=YC*SIN(W)+XO                                          ACSRC031
      YP(YO,YC,W)=YC*COS(W)+YO                                          ACSRC032
      DAYS=NODAYS                                                       ACSRC033
      NT=NPLTS+5                                                        ACSRC034
      IWIND=29+IWD                                                      ACSRC035
C                                                                       ACSRC036
C     AN IFLAG OF 0 MEANS THAT ALL AIRCRAFT SOURCES EXCEPT              ACSRC037
C     FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 REMAIN                 ACSRC038
C     UNCHANGED                                                         ACSRC039
C                                                                       ACSRC040
      IF(IFLAG.EQ.0) GO TO 69                                           ACSRC041
C                                                                       ACSRC042
C     READ ARRFCN AND DEPFCN FROM TAPE                                  ACSRC043
C                                                                       ACSRC044
      IF (IWD.GE.1.AND.IWD.LE.MAXFIL) GO TO 1000                        ACSRC045
      PRINT 9000,MRECPT,MAXFIL,IWD                                      ACSRC046
 9000 FORMAT(29HOFILE REQUEST ERROR IN ACSRCE,3I5)                      ACSRC047
      GO TO 1040                                                        ACSRC048
 1000 IF (MRECPT-IWD) 1010,1030,1020                                    ACSRC049
 1010 READ (30)                                                         ACSRC050
      MRECPT=MRECPT+1                                                   ACSRC051
      GO TO 1000                                                        ACSRC052
 1020 REWIND 30                                                         ACSRC053
      MRECPT=1                                                          ACSRC054
      GO TO 1000                                                        ACSRC055
 1030 READ (30) ARRFCN,DEPFCN                                           ACSRC056
      MRECPT=MRECPT+1                                                   ACSRC057
 1040 CONTINUE                                                          ACSRC058
C                                                                       ACSRC059
C     FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL                ACSRC060
C     DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY                         ACSRC061
```

```
C                                                                       ACSRC062
      NHI=IHR2                                                          ACSRC063
      IF(IHR1.GT.IHR2) NHI=24+IHR2                                      ACSRC064
      HRS=NHI-IHR1+1                                                    ACSRC065
      DO 5 I=1,NACTYP                                                   ACSRC066
      HRFRC=0.                                                          ACSRC067
      DO 4 JJ=IHR1,NHI                                                  ACSRC068
      J=JJ                                                              ACSRC069
      IF(JJ.GT.24) J=JJ-24                                              ACSRC070
 4    HRFRC=HRFRC+ACHR(J,I)                                             ACSRC071
      HRFRC=HRFRC/HRS                                                   ACSRC072
      FRAC(I)=ACMO(IMO,I)*ACDY(IDY,I)*HRFRC*7.0/DAYS*(1.E+6/3.6)        ACSRC073
 5    CONTINUE                                                          ACSRC074
    8 NACPT=0                                                           ACSRC075
      NB=0                                                              ACSRC076
      NC=0                                                              ACSRC077
      NZ=0                                                              ACSRC078
C                                                                       ACSRC079
C     SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS          ACSRC080
C                                                                       ACSRC081
      DO 1  L=1,NPKAR                                                   ACSRC082
      NSQ=NPASQ(L)                                                      ACSRC083
      SPARSQ=0.0                                                        ACSRC084
      DO 2 J=1,NSQ                                                      ACSRC085
      NB=NB+1                                                           ACSRC086
      ACAR(1,NB)=PAREA(L,J,1)                                           ACSRC087
      ACAR(2,NB)=PAREA(L,J,2)                                           ACSRC088
      ACAR(3,NB)=ACLNDZ/2.                                              ACSRC089
      ACAR(4,NB)=PAREA(L,J,3) *1000.                                    ACSRC090
      APARSQ(L,J)=ACAR(4,NB) ** 2                                       ACSRC091
      SPARSQ = SPARSQ + APARSQ(L,J)                                     ACSRC092
      ACAR(5,NB)=ACLNDZ                                                 ACSRC093
      IACAR(1,NB)=IDPRKA(L)                                             ACSRC094
  2   IACAR(2,NB)=NSQ                                                   ACSRC095
      DO 91 J=1,NSQ                                                     ACSRC096
      NZ=NZ+1                                                           ACSRC097
   91 PAFFCT(NZ) = APARSQ(L,J) / SPARSQ                                 ACSRC098
  1   CONTINUE                                                          ACSRC099
C                                                                       ACSRC100
      DO 93 I=1,NLSEGS                                                  ACSRC101
   93 NQ(I)=0                                                           ACSRC102
      NPKSRC=NB                                                         ACSRC103
      DO 3  L=1,NPKSRC                                                  ACSRC104
      DO 3  K=6,NT                                                      ACSRC105
      HRACAR(K-5,L)=0.0                                                 ACSRC106
  3   ACAR(K,L)=0.0                                                     ACSRC107
      TVP= EXP(ALPHA(2)-BETA(2)/TEMK)                                   ACSRC108
C                                                                       ACSRC109
C     BEGIN LOOP OVER N RUNWAYS                                         ACSRC110
C                                                                       ACSRC111
      DO 10 N=1,NRNWYS                                                  ACSRC112
C                                                                       ACSRC113
C     IS RUNWAY USED WITH THIS WIND DIRECTION?                          ACSRC114
C                                                                       ACSRC115
      IF (IUSWD(IWD,N).EQ.0) GO TO 10                                   ACSRC116
      THETA=RNWY(7,N)                                                   ACSRC117
      XO=0.25*DISRNW(N)*SIN(THETA)+RNWY(2,N)                            ACSRC118
      YO=0.25*DISRNW(N)*COS(THETA)+RNWY(3,N)                            ACSRC119
      NTT=NIBTT(N)                                                      ACSRC120
      IF(NTT.EQ.0) GO TO 50                                             ACSRC121
C                                                                       ACSRC122
C     BEGIN LOOP OVER J INBOUND TAXIWAYS                                ACSRC123
```

```
C                                                                       ACSRC124
      DO 11 J=1,NTT                                                     ACSRC125
C                                                                       ACSRC126
C     ANY AIRCRAFT ARRIVING ON THIS RUNWAY?                             ACSRC127
C                                                                       ACSRC128
      DO 7 I=1,NACTYP                                                   ACSRC129
      IF(TTARFR(J,I,N)*ARRFCN(23,I,N).GT.0.0) GO TO 701                 ACSRC130
    7 CONTINUE                                                          ACSRC131
      GO TO 11                                                          ACSRC132
  701 NSGLNS = NIBSEG (J,N)                                             ACSRC133
C                                                                       ACSRC134
C     BEGIN LOOP OVER K TAXIWAY SEGMENTS                                ACSRC135
C                                                                       ACSRC136
      DO 12 K=1,NSGLNS                                                  ACSRC137
C                                                                       ACSRC138
C     SET UP SEGMENT LINE SOURCE GEOMETRIES                             ACSRC139
C                                                                       ACSRC140
      JJ = IIBSEG(K,J,N)                                                ACSRC141
      IF(NQ(JJ).NE.0) GO TO 130                                         ACSRC142
      NC=NC+1                                                           ACSRC143
      NQ(JJ)=NC                                                         ACSRC144
      DO 121 L=1,12                                                     ACSRC145
  121 ACLN(L,NC)=ACLNSG(L,JJ)                                           ACSRC146
      ACLN(9,NC)=1.0                                                    ACSRC147
      ACLN(10,NC)=1.0                                                   ACSRC148
C                                                                       ACSRC149
C     ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS             ACSRC150
C     TO APPROPRIATE SEGMENTS                                           ACSRC151
C                                                                       ACSRC152
      DO 13 L=1,NPLTS                                                   ACSRC153
      LL=L+12                                                           ACSRC154
 13   ACLN(LL,NC)=0.0                                                   ACSRC155
  130 ND=NQ(JJ)                                                         ACSRC156
      DO 14 I=1,NACTYP                                                  ACSRC157
      AA=ENGNO(I ,1)                                                    ACSRC158
      IF(IEGFLG.GT.0) AA=ENGNO(I ,2)                                    ACSRC159
      ARR=TTARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)                        ACSRC160
      IF(ARR.LE.0.0)GO TO 14                                            ACSRC161
      TIME=ACLN(11,ND)/(TXISPD(I )*ACLNSG(9,JJ))                        ACSRC162
      FRC=AA*ARR*TIME*FRAC(I )                                          ACSRC163
      DO 15 L=1,NPLTS                                                   ACSRC164
      KK=L+12                                                           ACSRC165
 15   ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I ,2,L)                        ACSRC166
 14   CONTINUE                                                          ACSRC167
  12  CONTINUE                                                          ACSRC168
C                                                                       ACSRC169
C     END TAXIWAY SEGMENT LOOP                                          ACSRC170
C                                                                       ACSRC171
C                                                                       ACSRC172
C     DETERMINE AIRCRAFT INBOUND PARKING AREA                           ACSRC173
C     ASSOCIATED WITH TAXIWAY PATH                                      ACSRC174
C                                                                       ACSRC175
      DO 16 I=1,NPKSRC                                                  ACSRC176
      II=I                                                              ACSRC177
      IDPK=IACAR(1,I)                                                   ACSRC178
      IF(IDPK.EQ.IDIBPA(J,N))GO TO 17                                   ACSRC179
  16  CONTINUE                                                          ACSRC180
      PRINT 18, IDIBPA(J,N),J,N                                         ACSRC181
 18   FORMAT ('0INBOUND PARKING AREA 'I3,'OF TAXIWAY='I3,'; RUNWAY='I3,'ACSRC182
     1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')                 ACSRC183
      STOP                                                              ACSRC184
 17   CONTINUE                                                          ACSRC185
```

```
C
C     ALLOCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING,
C     ARRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS
C     TO APPROPRIATE AREA
C
      NSQ=IACAR(2,II)
      DO  19 I=1,NACTYP
      ARR=TTARRR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
      IF(ARR.LE.0.0)  GO TO 19
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0)  AA=ENGNO(I,2)
      TIME=SHTDNT(I)/60.
      FRC=AA*ARR*TIME*FRAC(I)
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)
      DO 20 L=1,NSQ
      JJ=II+L-1
      DO 21 K=1,NPLTS
      KK=K+5
      ACAR(KK,JJ)=ACAR(KK,JJ)+FRC*ACEMFC(I,1,K) * PARFCT(JJ)
      ACAR(KK,JJ)=ACAR(KK,JJ) + (ARSVEM(K,I,1) + ARSVEM(K,I,2) +
     .ARSVEM(K,I,3) + ARSVEM(K,I,4)+ARSVEM(K,I,5)) * ARR * FRAC(I)
     . * PARFCT(JJ)
      IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+(0.3*TVP*ACFUEL(I)*0.5
     1/1000. + ACSPII(I) + ARFLVT(I)) * ARR * FLDENS(JES1(I)) * FRAC(I)
     . * PARFCT(JJ)
   21 CONTINUE
   20 CONTINUE
   19 CONTINUE
   11 CONTINUE
C
C     END INBOUND TAXIWAY LOOP
C
C
C     BEGIN LOOP OVER I AIRCRAFT USED
C
      DO 30 I=1,NACTYP
C
C     CALCULATE RUNWAY ARRIVALS FOR EACH AIRCRAFT TYPE
C
      ARR=ARRFCN(23,I,N)*ANNARR(I)
C
C     ANY AIRCRAFT ARRIVING?
C
      IF(ARR.LE.0.0)  GO TO 30
C
C     SET UP LINE SOURCES FOR APPROACH AND LANDING MODES 7, 8 AND 9
C     AND ALLOCATE POLLUTANT EMISSIONS
C
      AA=ENGNO(I,1)
      DO 31 J=1,3
      DO 32 K=1,3
      KK=K+NC
      JK=6*K-6+J
      ACLN(J,KK)=ARRFCN(JK,I,N)
      ACLN(J+5,KK)=ARRFCN(JK+6,I,N)
   32 CONTINUE
      JJ=NC+J
      JK=6*J-2
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      ACLN(09,JJ)=ARRFCN(JK,I,N)
      ACLN(10,JJ)=ARRFCN(JK+6,I,N)
      ACLN(11,JJ)=ARRFCN(JK+1,I,N)                                      ACSRC248
      ACLN(12,JJ)=ARRFCN(JK+2,I,N)                                      ACSRC249
      JMODE=J+6                                                         ACSRC250
      DO 33 K=1,NPLTS                                                   ACSRC251
      KK=K+12                                                           ACSRC252
      ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*ARR*ARRFCN(JK+2,I,N)*FRAC(I)     ACSRC253
   33 CONTINUE                                                          ACSRC254
   31 CONTINUE                                                          ACSRC255
      NC=NC+3                                                           ACSRC256
C                                                                       ACSRC257
C     SET UP LINE SOURCES FOR TRAINING FLIGHT OPERATIONS                ACSRC258
C     AND ALLOCATE POLLUTANT EMISSIONS                                  ACSRC259
C                                                                       ACSRC260
      IF(ANNTGO(I).LE.0.0) GO TO 30                                     ACSRC261
      NC=NC+1                                                           ACSRC262
      ACLN(1,NC)=XP(XO,TGO(1,1,I),THETA)                                ACSRC263
      ACLN(2,NC)=YP(YO,TGO(1,1,I),THETA)                                ACSRC264
      ACLN(6,NC)=XP(XO,TGO(1,2,I),THETA)                                ACSRC265
      ACLN(7,NC)=YP(YO,TGO(1,2,I),THETA)                                ACSRC266
      ACLN(1,NC+1)=ACLN(6,NC)                                           ACSRC267
      ACLN(2,NC+1)=ACLN(7,NC)                                           ACSRC268
      ACLN(6,NC+1)=XO                                                   ACSRC269
      ACLN(7,NC+1)=YO                                                   ACSRC270
      ACLN(1,NC+2)=XO                                                   ACSRC271
      ACLN(2,NC+2)=YO                                                   ACSRC272
      ACLN(6,NC+2)=XP(XO,0.3048,THETA)                                  ACSRC273
      ACLN(7,NC+2)=YP(YO,0.3048,THETA)                                  ACSRC274
      ACLN(1,NC+3)=ACLN(6,NC+2)                                         ACSRC275
      ACLN(2,NC+3)=ACLN(7,NC+2)                                         ACSRC276
      ACLN(6,NC+3)=XP(XO,TGO(1,3,I),THETA)                              ACSRC277
      ACLN(7,NC+3)=YP(YO,TGO(1,3,I),THETA)                              ACSRC278
      ACLN(1,NC+4)=ACLN(6,NC+3)                                         ACSRC279
      ACLN(2,NC+4)=ACLN(7,NC+3)                                         ACSRC280
      ACLN(6,NC+4)=XP(XO,TGO(1,4,I),THETA)                              ACSRC281
      ACLN(7,NC+4)=YP(YO,TGO(1,4,I),THETA)                              ACSRC282
      ACLN(3,NC)=APPHT*1000.                                            ACSRC283
      ACLN(8,NC)=APPHT2(I)*1000.                                        ACSRC284
      ACLN(3,NC+1)=APPHT2(I)*1000.                                      ACSRC285
      ACLN(8,NC+1)=ACLNDZ/2.                                            ACSRC286
      ACLN(3,NC+2)=ACLNDZ/2.                                            ACSRC287
      ACLN(8,NC+2)=ACLNDZ/2.                                            ACSRC288
      ACLN(3,NC+3)=ACLNDZ/2.                                            ACSRC289
      ACLN(8,NC+3)=COHT1(I)*1000.                                       ACSRC290
      ACLN(3,NC+4)=COHT1(I)*1000.                                       ACSRC291
      ACLN(8,NC+4)=CLMBHT*1000.                                         ACSRC292
      ACLN(09,NC)=APSPD1(I)                                             ACSRC293
      ACLN(10,NC)=APSPD2(I)                                             ACSRC294
      ACLN(11,NC)=TGO(2,1,I)                                            ACSRC295
      ACLN(12,NC)=TGO(3,1,I)                                            ACSRC296
      ACLN(09,NC+1)=APSPD2(I)                                           ACSRC297
      ACLN(10,NC+1)=LNDSPD(I)                                           ACSRC298
      ACLN(11,NC+1)=TGO(2,2,I)                                          ACSRC299
      ACLN(12,NC+1)=TGO(3,2,I)                                          ACSRC300
      ACLN(09,NC+2)=LNDSPD(I)*1.3                                       ACSRC301
      ACLN(10,NC+2)=TOSPD(I)*0.7                                        ACSRC302
      ACLN(11,NC+2)=0.3048                                              ACSRC303
      ACLN(12,NC+2)=2.0*0.3048/(1.3*LNDSPD(I)+0.7*TOSPD(I))             ACSRC304
      ACLN(09,NC+3)=TOSPD(I)                                            ACSRC305
      ACLN(10,NC+3)=COSPD1(I)                                           ACSRC306
      ACLN(11,NC+3)=TGO(2,3,I)                                          ACSRC307
      ACLN(12,NC+3)=TGO(3,3,I)                                          ACSRC308
      ACLN(09,NC+4)=COSPD1(I)                                           ACSRC309
      ACLN(10,NC+4)=COSPD2(I)
      ACLN(11,NC+4)=TGO(2,4,I)
      ACLN(12,NC+4)=TGO(3,4,I)
      DO 45 J=1,5
      JJ=NC+J-1
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      GO TO (34,35,41,36,37),J
 34   KD=7
      GO TO 38
 35   KD=8
      GO TC 38
 36   KD=5
      GO TO 38
 37   KD=6
 38   DO 39 K=1,NPLTS
      KK=K+12
 39   ACLN(KK,JJ)=ANNTGO(I)*ACEMFC(I,KD,K)*ARRFCN(23,I,N)*ACLN(12,JJ)*
     1FRAC(I)*AA
      GO TO 45
 41   DO 42 K=1,NPLTS
      KK=K+12
 42   ACLN(KK,JJ)=AA*(0.3*ACEMFC(I,9,K)+0.7*ACEMFC(I,4,K))*
     1 ANNTGO(I)*ARRFCN(23,I,N)*ACLN(12,JJ)*FRAC(I)
 45   CONTINUE
      NC=NC+4
   30 CONTINUE
C
C     END AIRCRAFT LOOP
C
 50   NTT=NOBTT(N)
      IF(NTT.EQ.0) GO TO 10
C
C     BEGIN LOOP OVER J OUTBOUND TAXIWAYS
C
      DO 51 J=1,NTT
C
C     ANY AIRCRAFT DEPARTING ON THIS TAXIWAY?
C
      DO 6 I=1,NACTYP
      IF(TTDPFR(J,I,N)*DEPFCN(23,I,N).GT.0.0) GO TO 601
    6 CONTINUE
      GO TO 51
  601 NSGLNS=NOBSEG(J,N)
C
C     BEGIN LOOP OVER K TAXIWAY SEGMENTS
C
      DO 52 K=1,NSGLNS
C
C     SET UP SEGMENT LINE SOURCE GEOMETRIES
C
      JJ=IOBSEG(K,J,N)
      IF(NQ(JJ).NE.0) GO TO 131
      NC=NC+1
      NQ(JJ)=NC
      DO 122 L=1,12
  122 ACLN(L,NC)=ACLNSG(L,JJ)
      ACLN(9,NC)=1.0
      ACLN(10,NC)=1.0
C
C     ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS
C     TO APPROPRIATE SEGMENTS
```

```
C
      DO 53 L=1,NPLTS
      LL=L+12
   53 ACLN(LL,NC)=0.0
  131 ND=NC(JJ)
      DO 54 I=1,NACTYP
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.LE.0.0) GO TO 54
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME= ACLN(11,ND) / TXISPD(I)
      FRC= AA* DEP*TIME*FRAC(I)
      DO 55 L=1,NPLTS
      KK=L+12
   55 ACLN(KK,ND)= ACLN(KK,ND)+FRC*ACEMFC(I,2,L)
   54 CONTINUE
   52 CONTINUE
C
C     END TAXIWAY SEGMENT LOOP
C
C
C     DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED
C     WITH TAXIWAY PATH
C
      DO 56 I=1,NPKSRC
      II =I
      IDPK=IACAR(1,I)
      IF(IDPK.EQ.IDOBPA(J,N)) GO TO 58
   56 CONTINUE
      PRINT 57,IDOBPA(J,N),J,N
   57 FORMAT(22H0OUTBOUND PARKING AREA,I3,11H OF TAXIWAY,I3,8H, RUNWAY,
     . I3,47H IS NOT CONSISTENT WITH PARKING AREA ID NUMBERS)
      STOP
C
C     ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL
C     VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
C
   58 NSQ=IACAR(2,II )
      DO 59 I=1,NACTYP
      DEP=TTDPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.EQ.0.0) GO TO 59
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME=SRTUPT(I)/60.
      FRC = AA* DEP* TIME * FRAC(I)
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)
      DO 60 L=1,NSQ
      JJ =II +L-1
      DO 61 K=1,NPLTS
      KK=K+5
      ACAR(KK,JJ) = ACAR(KK,JJ) + ((FRC * ACEMFC(I,1,K)) -
     . ((DPSVEM(K,I,1) + DPSVEM(K,I,2) + DPSVEM(K,I,3) + DPSVEM(K,I,4)
     . + DPSVEM(K,I,5)) * DEP * FRAC(I))) * PARFCT(JJ)
      IF (K.EQ.2) ACAR(KK,JJ) = ACAR(KK,JJ) + DPFLVT(I) * DEP * FLDENS(
     . JES1(I)) * FRAC(I) * PARFCT(JJ)
   61 CONTINUE
   60 CONTINUE
   59 CONTINUE
   51 CONTINUE
C
C     END OUTBOUND TAXIWAY LOOP
C
```

```
      NB=NB+1                                                           ACSRC434
C                                                                       ACSRC435
C     SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE                 ACSRC436
C     ENGINE CHECK EMISSIONS TO IT                                      ACSRC437
C                                                                       ACSRC438
      ACAR(1,NB)=RNWY(2,N)-.05 * SIN(THETA)                             ACSRC439
      ACAR(2,NB)=RNWY(3,N)-.05*COS(THETA)                               ACSRC440
      ACAR(3,NB)= ACINDZ/2.                                             ACSRC441
      ACAR(4,NB)= 100.0                                                 ACSRC442
      ACAR(5,NB)= ACINDZ                                                ACSRC443
      DO 65 K=1,NPLTS                                                   ACSRC444
      KK=K+5                                                            ACSRC445
   65 ACAR(KK,NB)=0.0                                                   ACSRC446
      DO 66 I=1,NACTYP                                                  ACSRC447
      DEP=DEPFCN(23,I,N)*ANNDEP(I)                                      ACSRC448
      IF(DEP.EQ.0.0)  GO TO 66                                          ACSRC449
      AA=ENGNO(I,1)                                                     ACSRC450
      IF(IEGFLG.GT.0)  AA=ENGNO(I,2)                                    ACSRC451
      TIME= EGCHKT(I)/60.                                               ACSRC452
      FRC= TIME *DEP*AA*FRAC(I)                                         ACSRC453
      DO 67 K=1,NPLTS                                                   ACSRC454
      KK=K+5                                                            ACSRC455
   67 ACAR(KK,NB)=ACAR(KK,NB)+ FRC* ACEMFC(I,3,K)                       ACSRC456
   66 CONTINUE                                                          ACSRC457
   10 CONTINUE                                                          ACSRC458
C                                                                       ACSRC459
C     END RUNWAY LOOP                                                   ACSRC460
C                                                                       ACSRC461
      NACAR=NB                                                          ACSRC462
      NC1=NC                                                            ACSRC463
   69 NC=NC1                                                            ACSRC464
C                                                                       ACSRC465
C     BEGIN LOOP OVER N RUNWAYS                                         ACSRC466
C                                                                       ACSRC467
      DO 79 N=1,NRNWYS                                                  ACSRC468
C                                                                       ACSRC469
C     IS RUNWAY USED WITH THIS WIND DIRECTION?                          ACSRC470
C                                                                       ACSRC471
      IF(IUSWD(IWD,N).EQ.0)  GO TO 79                                   ACSRC472
C                                                                       ACSRC473
C     BEGIN LOOP OVER I AIRCRAFT USED                                   ACSRC474
C                                                                       ACSRC475
      DO 70 I=1,NACTYP                                                  ACSRC476
C                                                                       ACSRC477
C     CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE                ACSRC478
C                                                                       ACSRC479
      DEP=DEPFCN(23,I,N)*ANNDEP(I)                                      ACSRC480
C                                                                       ACSRC481
C     ANY AIRCRAFT DEPARTING FROM THIS RUNWAY?                          ACSRC482
C                                                                       ACSRC483
      IF(DEP.EQ.0.0)  GO TO 70                                          ACSRC484
C                                                                       ACSRC485
C     CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING         ACSRC486
C     TO CURRENT METEOROLOGICAL CONDITIONS                              ACSRC487
C                                                                       ACSRC488
      CALL DEPART  (N,I)                                                ACSRC489
      AA=ENGNO(I  ,1)                                                   ACSRC490
C                                                                       ACSRC491
C     SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2    ACSRC492
C     AND ALLOCATE POLLUTANT EMISSIONS                                  ACSRC493
C                                                                       ACSRC494
      DO 71 J=1,3                                                       ACSRC495
      DO 72 K=1,3
      KK=K+NC
      JK=6*K-6+J
      ACLN(J,KK)=DEPFCN(JK,I,N)
   72 ACLN(J+5,KK)=DEPFCN(JK+6,I,N)
      JJ=NC+J
      JK=6*J-2
      ACLN(4,JJ)=ARRFCN(24,I,N)
      ACLN(5,JJ)=DEPFCN(24,I,N)
      ACLN(09,JJ)=DEPFCN(JK,I,N)
      ACLN(10,JJ)=DEPFCN(JK+6,I,N)
      ACLN(11,JJ)=DEPFCN(JK+1,I,N)
      ACLN(12,JJ)=DEPFCN(JK+2,I,N)
      JMODE=J+3
      DO 73 K=1,NPLTS
      KK=K+12
      ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*DEP*DEPFCN(JK+2,I,N)*FRAC(I)
   73 CONTINUE
   71 CONTINUE
      NC=NC+3
   70 CONTINUE
C
C     END AIRCRAFT LOOP
C
   79 CONTINUE
C
C     END RUNWAY LOOP
C
      NACLN=NC
      RETURN
      END
```

FUNCTION AINE

Purpose:

1. To translate the line and receptor coordinates to an x-axis along the wind vector, placing the origin of the line at its low end.
2. To set up the necessary parameters for the CAVL and QMOD routines.
3. To determine the concentration due to the given line.

Input:

The current wind direction and speed, and the receptor and line source data.

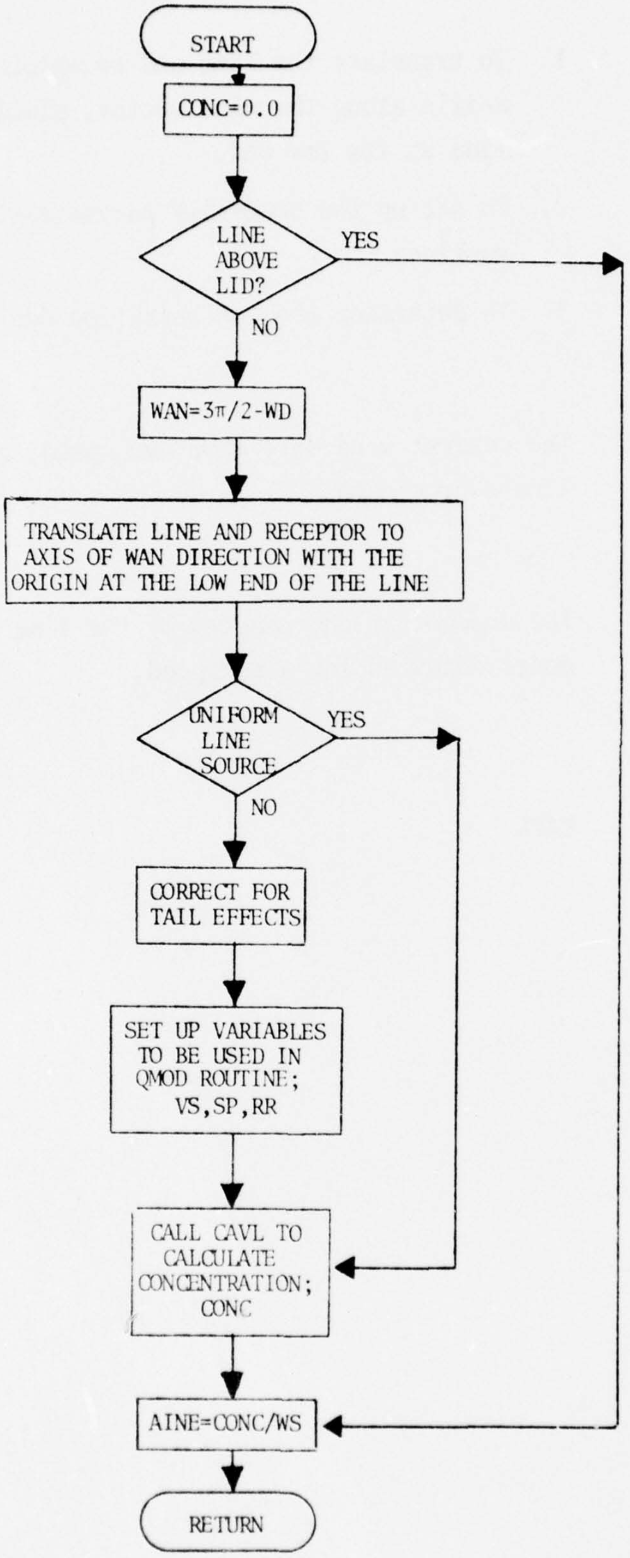
Output:

The concentration computed by the line source diffusion model adjusted for wind speed.

Subroutines Called:

CAVL

FUNCTION AINE(WD)


START
CONC=0.0
LINE ABOVE LID?
YES
NO
WAN=3π/2-WD
TRANSLATE LINE AND RECEPTOR TO AXIS OF WAN DIRECTION WITH THE ORIGIN AT THE LOW END OF THE LINE
UNIFORM LINE SOURCE
YES
NO
CORRECT FOR TAIL EFFECTS
SET UP VARIABLES TO BE USED IN QMOD ROUTINE; VS,SP,RR
CALL CAVL TO CALCULATE CONCENTRATION; CONC
AINE=CONC/WS
RETURN

```
      FUNCTION AINE (WD)
C
C   THIS FUNCTION TRANSLATES THE LINE AND RECEPTOR COORDINATES TO AN
C   X-AXIS ALONG THE WIND VECTOR, PLACING THE ORIGIN OF THE LINE AT
C   ITS LOW END.  THE VEHICLE MOVES FROM (X1,Y1,Z1) TO (X2,Y2,Z2)
C
      COMMON /MET/ WS,WSMPH,IWS,WX,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND
      DATA PI32/4.7123890/
C
      CONC = 0.
C
C     IF LINE IS ABOVE LID, DO NOT CALCULATE CONC
C
      IF(ZW1.GE.HLID-.5) GO TO 60
C
C     TRANSLATE LINE AND RECEPTOR TO AXIS OF WAN DIRECTION
C
      WAN=PI32-WD
      CSAN=COS(WAN)
      SNAN=SIN(WAN)
      XW2=(X2-X1)*CSAN+(Y2-Y1)*SNAN
      YW2=(X1-X2)*SNAN+(Y2-Y1)*CSAN
      XR = RECEP(1,IRECEP) * 1000.
      YR = RECEP(2,IRECEP) * 1000.
      ZST=ZW1
      ZND=ZW2
      IF(Z1.LE.Z2) GO TO 5
      XW2=-XW2
      XST=XW2
      YW2=-YW2
      YST=YW2
      XND=0.0
      YND=0.0
      XRCP=(XR-X2)*CSAN+(YR-Y2)*SNAN
      YRCP=(X2-XR)*SNAN+(YR-Y2)*CSAN
      GO TO 8
    5 CONTINUE
      XST=0.0
      YST=0.0
      XND=XW2
      YND=YW2
      XRCP=(XR-X1)*CSAN+(YR-Y1)*SNAN
      YRCP=(X1-XR)*SNAN+(YR-Y1)*CSAN
    8 CONTINUE
      ZRCP = 2.
C
C     IS THIS A UNIFORM LINE SOURCE
C
   50 IF(IAD.EQ.0) GO TO 500
C
C     CORRECT FOR TAIL EFFECTS IF ARRIVAL OR DEPARTURE
C
      CSA = -XW2 / DL
      WSC = 2 * WS * CSA
      EXT = TAIL / DL
      DX = XW2 * EXT
      DY = YW2 * EXT                                                    AINE0062
      XW2 = XW2 + DX                                                    AINE0063
      YW2 = YW2 + DY                                                    AINE0064
      VS = TAIL / TIME                                                  AINE0065
      W1 = V1 + VS                                                      AINE0066
      W2 = V2 + VS                                                      AINE0067
      YY1 = SQRT(WS2 + W1 * (W1 + WSC))                                 AINE0068
      YY2 = SQRT(WS2 + W2 * (W2 + WSC))                                 AINE0069
      SP = YY2                                                          AINE0070
      ARG = (YY2+ W2 + WSC/2.) / (YY1+ W1 + WSC/2.)                     AINE0071
      G =YY2 -YY1 - WSC/2. * ALOG(ARG)                                  AINE0072
      RR = A / G                                                        AINE0073
      IF (Z1.NE.Z2.AND.IAD.EQ.1) GO TO 500                              AINE0074
      XRCP = XRCP + DX                                                  AINE0075
      YRCP = YRCP + DY                                                  AINE0076
C                                                                       AINE0077
C   CALCULATE THE CONCENTRATION DUE TO THIS LINE                        AINE0078
C                                                                       AINE0079
  500 CONC=CAVL(XRCP,YRCP,ZRCP)                                         AINE0080
   60 AINE = CONC / WS                                                  AINE0081
      RETURN                                                            AINE0082
      END                                                               AINE0083
```

FUNCTION AREAWT

Purpose:

To calculate the area weighting factor to account for the fraction of the source seen by the receptor.

Input:

1. Distance from receptor to pseudo source origin.
2. Adjusted source to receptor distance measured from the center of mass of that part of the source seen.
3. Source diameter and radius.

Output:

The area weighting factor, AREAWT.

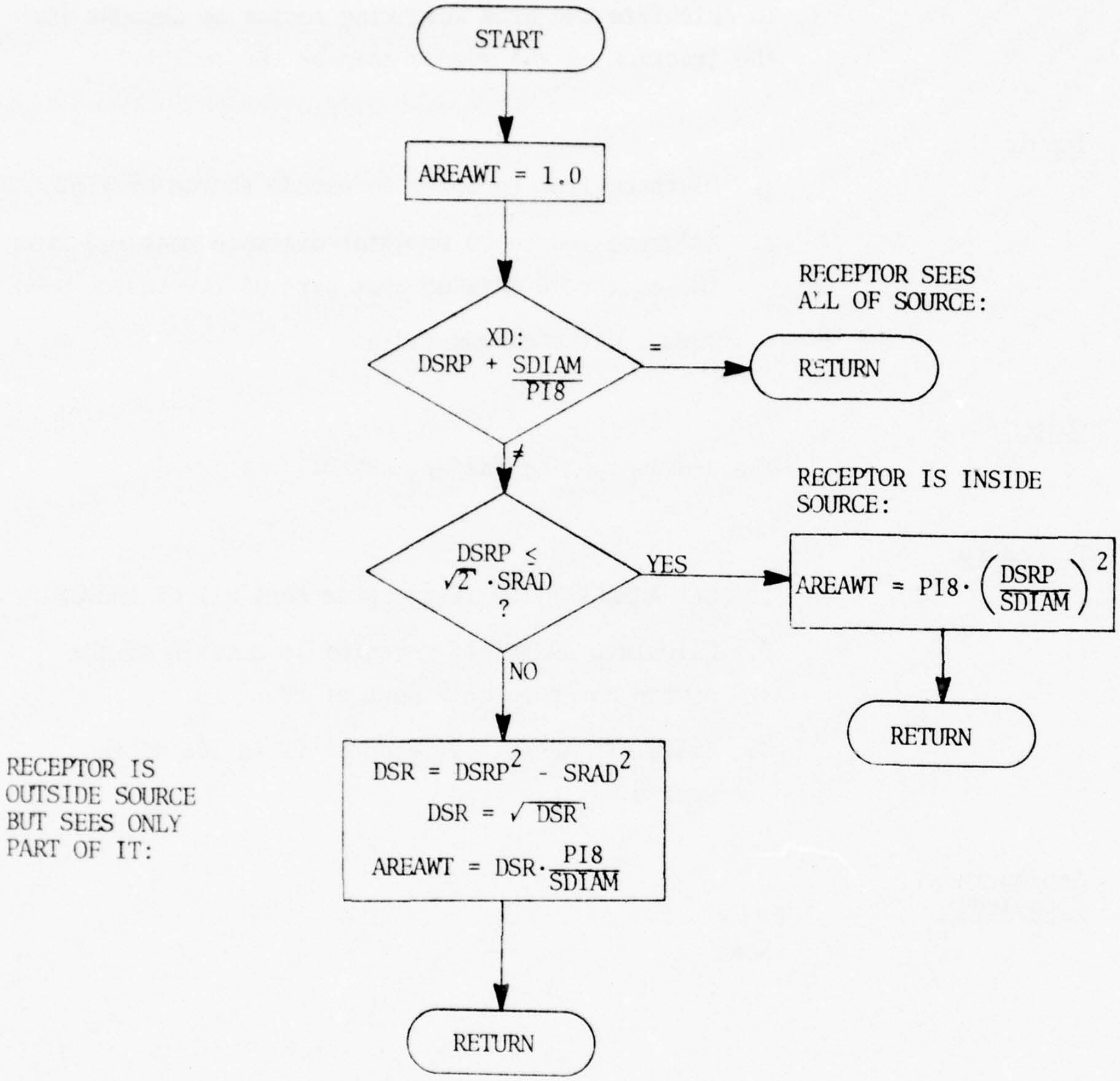
Procedure:

1. Set AREAWT = 1.0 if receptor sees all of source.
2. Calculate AREAWT if receptor is outside of the source but sees only part of it.
3. Calculate AREAWT if receptor is inside of the source.

Subroutines Called:

None

FUNCTION AREAWT(XD,DSRP,SDIAM, SRAD, PI8)


START
AREAWT = 1.0
XD:
DSRP + SDIAM / PI8
=
RECEPTOR SEES ALL OF SOURCE:
RETURN
≠
DSRP ≤ √2 · SRAD ?
YES
RECEPTOR IS INSIDE SOURCE:
AREAWT = PI8 · (DSRP / SDIAM)²
RETURN
NO
RECEPTOR IS OUTSIDE SOURCE BUT SEES ONLY PART OF IT:
DSR = DSRP² - SRAD²
DSR = √DSR
AREAWT = DSR · PI8 / SDIAM
RETURN

```
      FUNCTION AREAWT(XD,DSRP,SDIAM,SRAD,PI8)
C
C   THIS FUNCTION CALCULATES AREA WEIGHTING FACTOR TO ACCOUNT FOR THE
C   FRACTION OF THE SOURCE 'SEEN' BY THE RECEPTOR
C     XD IS DISTANCE FROM RECEPTOR TO PSEUDO SOURCE ORIGIN
C     DSRP IS ADJUSTED SOURCE-RECEPTOR DISTANCE MEASURED FROM
C     THE CENTER OF MASS OF THAT PART OF THE SOURCE 'SEEN'
C     SDIAM, SRAD, PI8 ARE SOURCE DIAMETER, RADIUS AND PI/8
C
      AREAWT=1.0
C
C   WEIGHTING FACTOR IS 1.0 IF RECEPTOR SEES ALL OF SOURCE
C
      IF(XD.EQ.DSRP+SDIAM/PI8) RETURN
      IF(DSRP.LE.1.41421356*SRAD) GO TO 10
C
C   RECEPTOR IS OUTSIDE SOURCE BUT SEES ONLY PART OF IT
C   AREA SEEN IS APPROXIMATED BY ASSUMING THE SECTOR CONE LIES
C   COMPLETELY WITHIN THE SOURCE SUCH THAT THIS AREA IS
C   GIVEN BY SDIAM*DSR*PI8 SO THE RATIO IS DSR*PI8/SDIAM WHERE
C   DSR IS THE ACTUAL SOURCE-RECEPTOR DISTANCE
C
      DSR=DSRP*DSRP-SRAD*SRAD
      DSR=SQRT(DSR)
      AREAWT=DSR*PI8/SDIAM
      RETURN
   10 CONTINUE
C
C   RECEPTOR IS INSIDE SOURCE, AREA SEEN IS .5*(DSR+SRAD)**2*PI8,
C   BUT DSR+SRAD=SQRT(2)*DSRP, SO AREA IS PI8*DSRP**2
C
      AREAWT=PI8*(DSRP/SDIAM)**2
      RETURN
      END
```

BLOCK DATA

Purpose:

To initialize data in common blocks.

Input:

None

Output:

None

```
      BLOCK DATA                                                        BLKDT000
C                                                                       BLKDT001
C   INITIALIZE DATA IN COMMON BLOCKS FOR LONG TERM MODEL                BLKDT002
C                                                                       BLKDT003
      REAL*8 POLNAM                                                     BLKDT004
      COMMON /ANNMET/ TBAR,ADD,PA,PAX,WSBAR,DTBAR                       BLKDT005
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)             BLKDT006
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7) BLKDT007
      COMMON /LN/ XW1 YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,B,V12,VS,BLKDT008
     . WS2,WSC,RR,SP,AA1,AA2,AA3,AA4,AA5,AA6                            BLKDT009
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)          BLKDT010
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,     BLKDT011
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),  BLKDT012
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)    BLKDT013
      COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,IP           BLKDT014
      COMMON /WNDPRO/ XP(6)                                             BLKDT015
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),       BLKDT016
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13), BLKDT017
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1         BLKDT018
C                                                                       BLKDT019
C     ***********************DATA STATEMENTS****************************BLKDT020
C                                                                       BLKDT021
      DATA PI4,PI8,PI16 /.7853982,.3926991,.1963496/                    BLKDT022
      DATA XW1 /0.0/,YW1 /0.0/ , TAIL /140./                            BLKDT023
      DATA XP/0.2,0.2,0.2,0.3,0.4,0.4/                                  BLKDT024
      DATA ALPHA /11.70365, 11.10675, 12.42382, 12.68789, 13.687,       BLKDT025
     . 13.038, 13.024 /                                                 BLKDT026
      DATA BETA / 2868.54, 3129.5187, 3276.8848, 5108.4194,5329.139,    BLKDT027
     . 4789.301, 4782.209 /                                             BLKDT028
      DATA FLDENS / 0.695, 0.773, 0.693, 0.842, 0.824, 0.807, 0.807 /   BLKDT029
      DATA ACLNDY, ACLNDZ / 20.0,8.0 /                                  BLKDT030
      DATA ITAPE / 21 /                                                 BLKDT031
      DATA POLNAM/8H   CO   ,8H   HC   ,8H  NOX   ,8H   PT   ,          BLKDT032
     .8H  SO2   ,8H  POL6  /                                            BLKDT033
      DATA ENPT,ENAR,ENLN,ABPT,ABAR,ABLN,ACPT,ACAR,ACLN /12660*0.0/     BLKDT034
      DATA NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,NACAR,NACLN/9*0.0/ BLKDT035
C                                                                       BLKDT036
C     COSWD(IWNDIR)=COS((PI*FLOAT(IWNDIR-1))/8)                         BLKDT037
C                                                                       BLKDT038
      DATA COSWD/1.0,.92388,.70711,.38268,0.0,-.38268,-.70711,-.92388,  BLKDT039
     1    -1.0,-.92388,-.70711,-.38268,0.0,.38268,.70711,.92388/        BLKDT040
C                                                                       BLKDT041
C     SINWD(IWNDIR)=SIN((PI*FLOAT(IWNDIR-1))/8)                         BLKDT042
C                                                                       BLKDT043
      DATA SINWD/0.0,.38268,.70711,.92388,1.0,.92388,.70711,.38268,0.0, BLKDT044
     1    -.38268,-.70711,-.92388,-1.0,-.92388,-.70711,-.38268/         BLKDT045
C                                                                       BLKDT046
C     WIND SPEED CORESPONDING TO AF WIND CLASSES                        BLKDT047
C                                                                       BLKDT048
      DATA UU /.77175,2.8297,5.145,7.9747,11.062,13.891/                BLKDT049
      END                                                               BLKDT050
```

FUNCTION CAVL

Purpose:

To compute the coupling coefficient at a receptor due to a line source.

Input:

Meteorological conditions: wind speed; stability; mixing height; critical distance for vertical mixing; psuedo downwind distances for horizontal and vertical spreads of the line source.

Source parameters: end point coordinates of the line (X-axis has been chosen to be along the wind vector); IAD flag for uniform or non-uniform line.

Receptor coordinates.

Output:

CAVL, the coupling coefficient.

Procedure:

1. Test whether the receptor is located with respect to the line source such that the concentration is negligible.
2. If the angle between the wind vector and line is sufficiently small, and the line is sufficiently long, set a flag for the line to be segmented.
3. Compute effective downwind distance and the horizontal and vertical dispersion coefficients.
4. Determine factor to be used in subdividing the line.
5. Test whether the line has a uniform density. If it is a runway used for aircraft arrival or departure (non-uniform density), call subroutine QMOD.
6. Determine the proper expression to be used and compute the concentration due to the line segment.
7. Test whether further segments need be considered. If not, output the concentration for the given receptor.

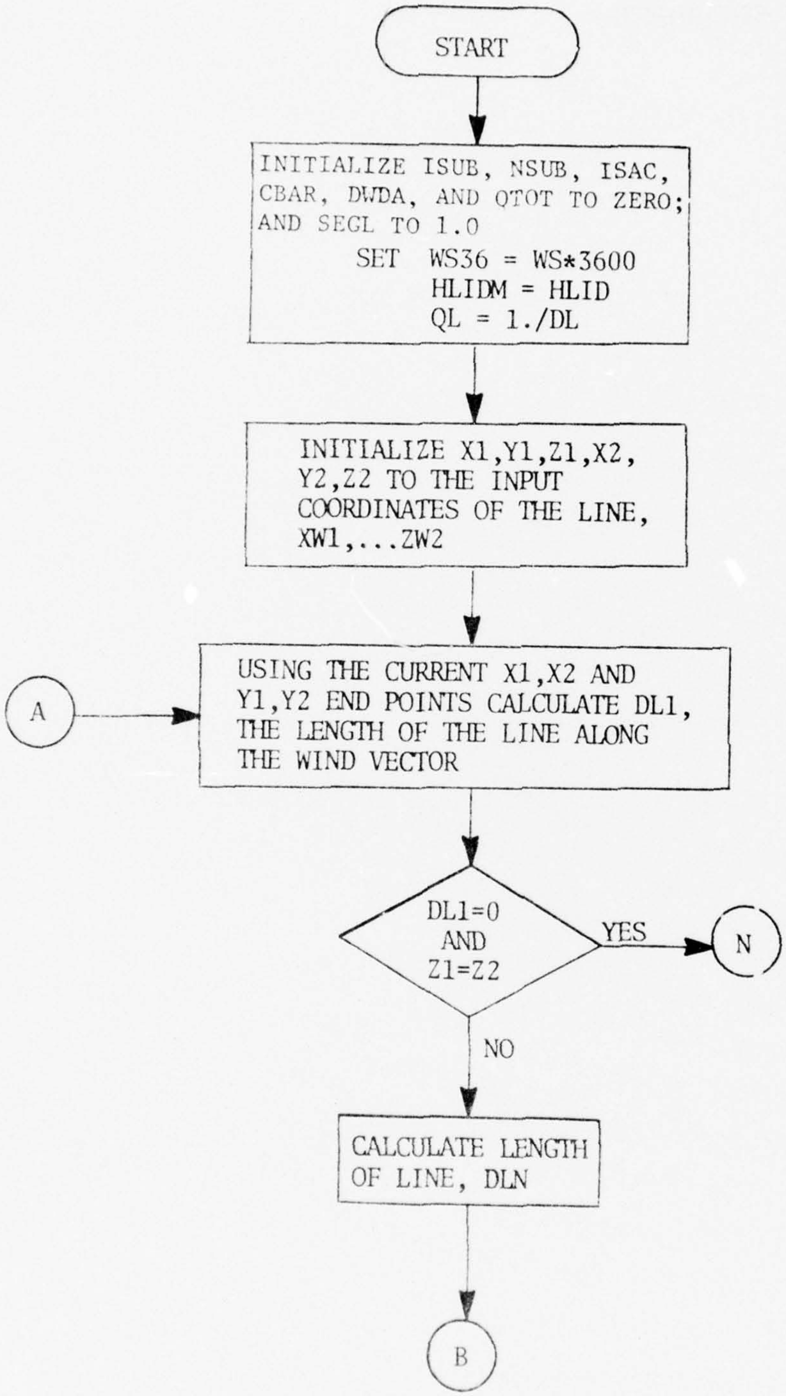
Functions <u>Called</u>:

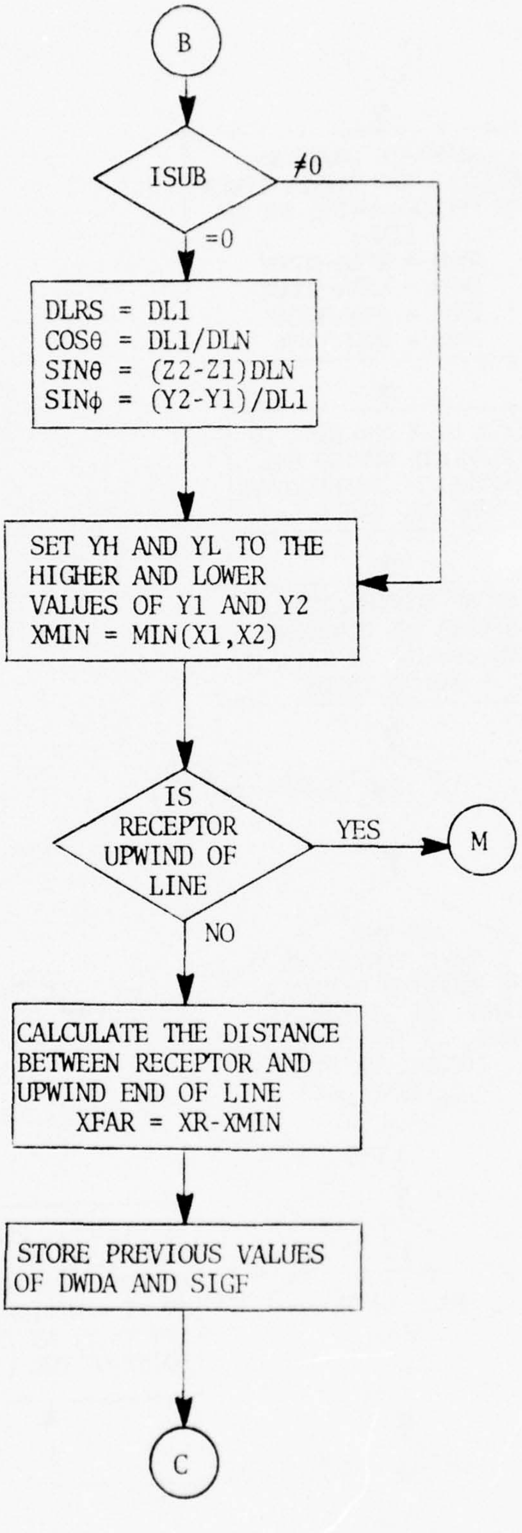
SIGY,SIGZ,DIFERF

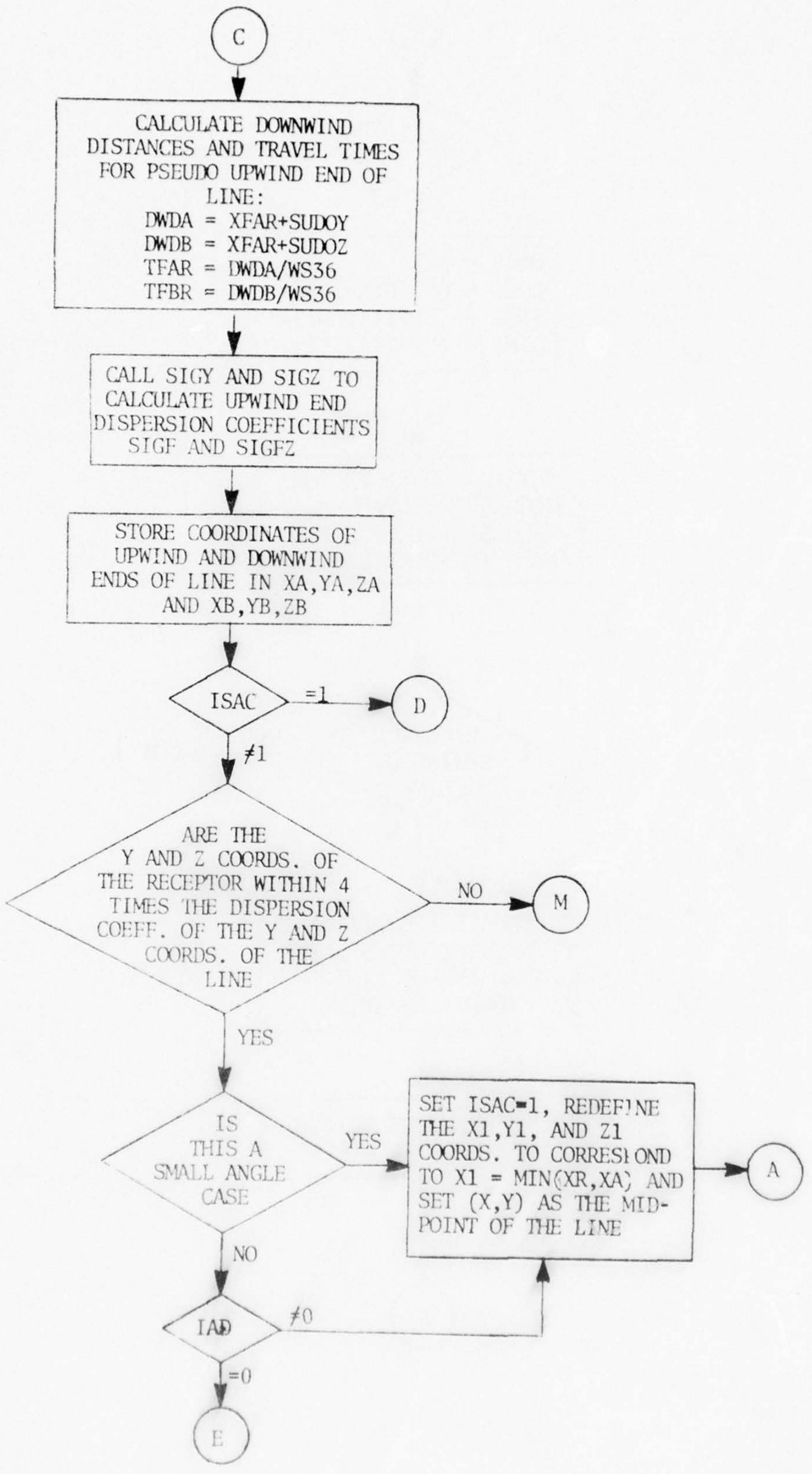
Subroutine <u>Called</u>:

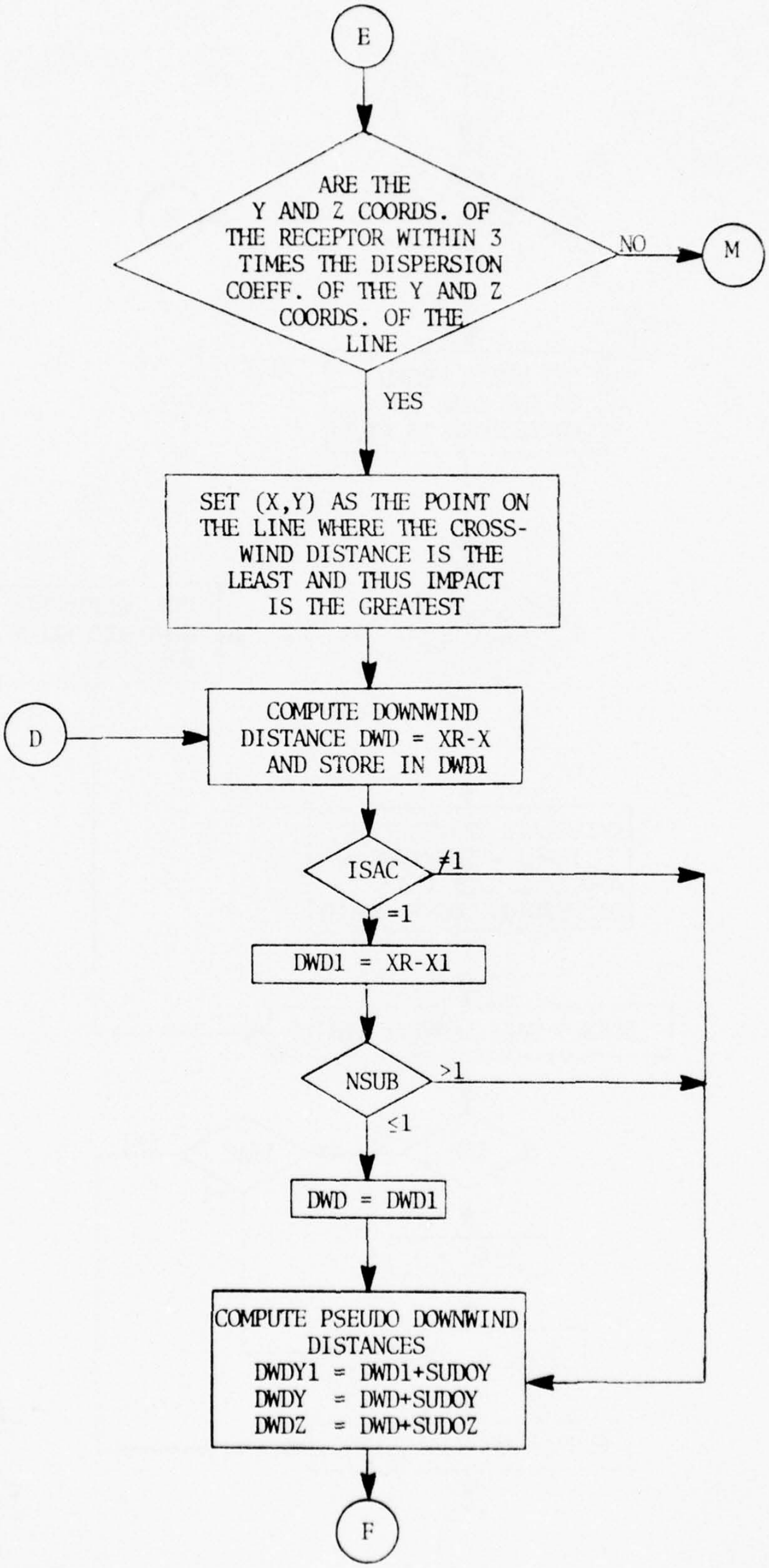
QMOD

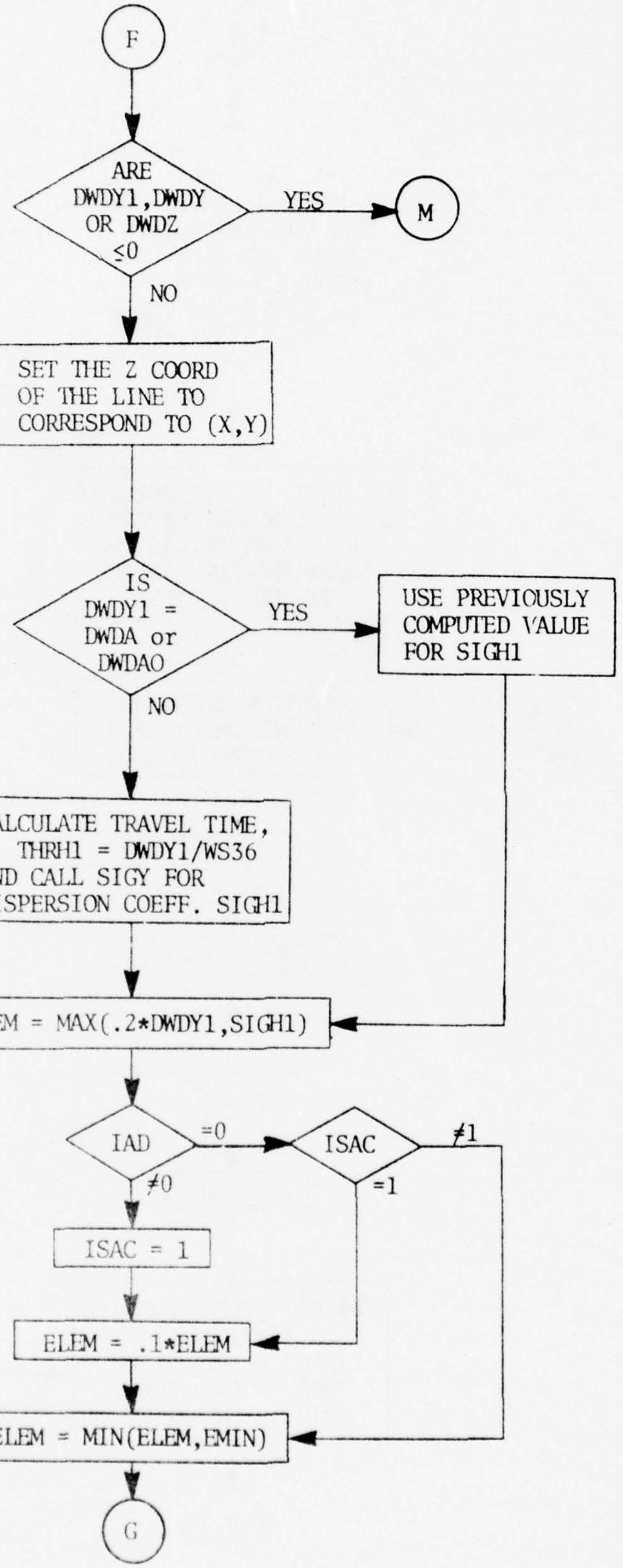
FUNCTION CAVL(XR,YR,ZR)

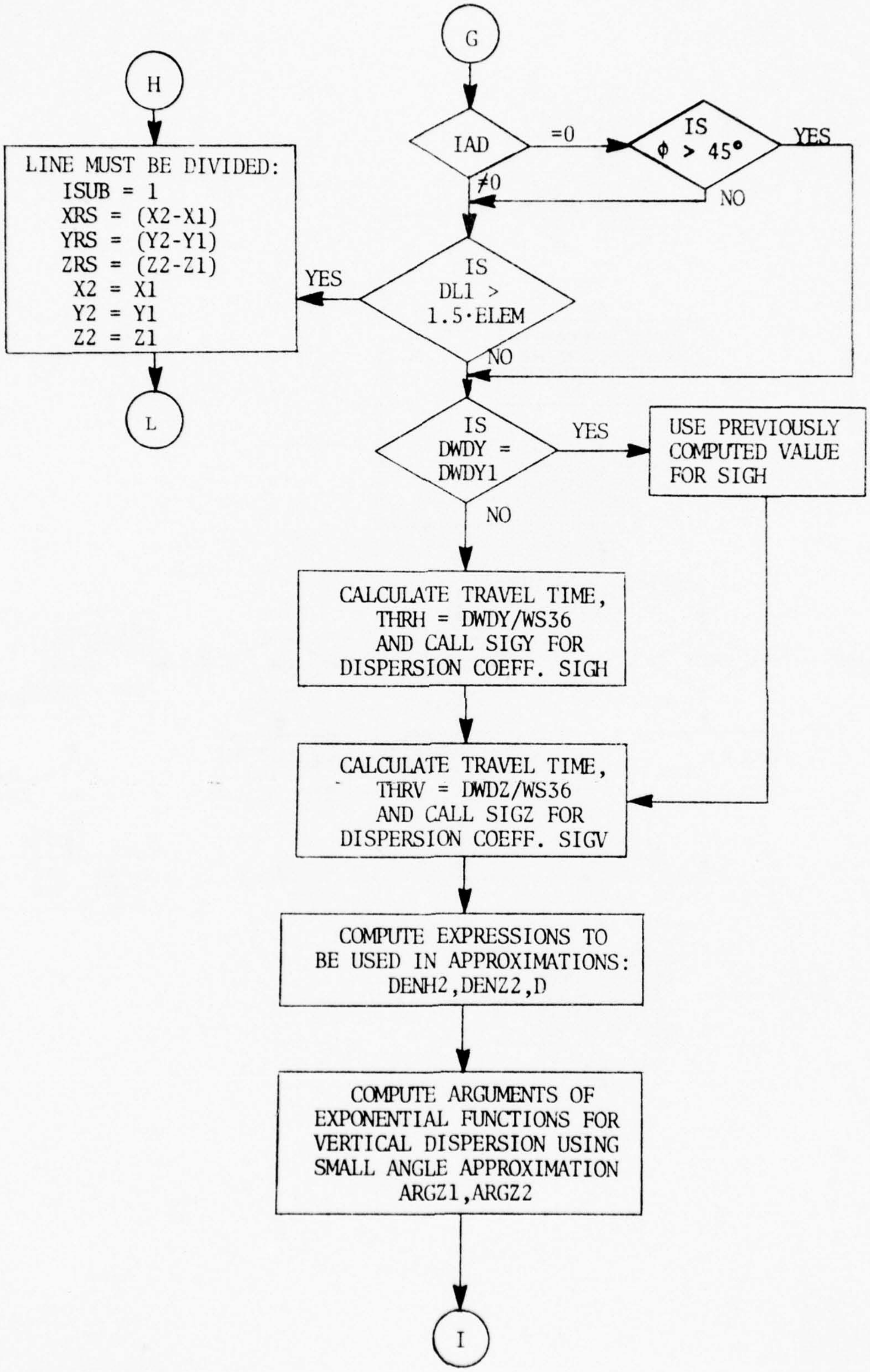

START
INITIALIZE ISUB, NSUB, ISAC, CBAR, DWDA, AND QTOT TO ZERO; AND SEGL TO 1.0
SET WS36 = WS*3600
HLIDM = HLID
QL = 1./DL
INITIALIZE X1,Y1,Z1,X2, Y2,Z2 TO THE INPUT COORDINATES OF THE LINE, XW1,...ZW2
A
USING THE CURRENT X1,X2 AND Y1,Y2 END POINTS CALCULATE DL1, THE LENGTH OF THE LINE ALONG THE WIND VECTOR
DL1=0 AND Z1=Z2
YES
N
NO
CALCULATE LENGTH OF LINE, DLN
B

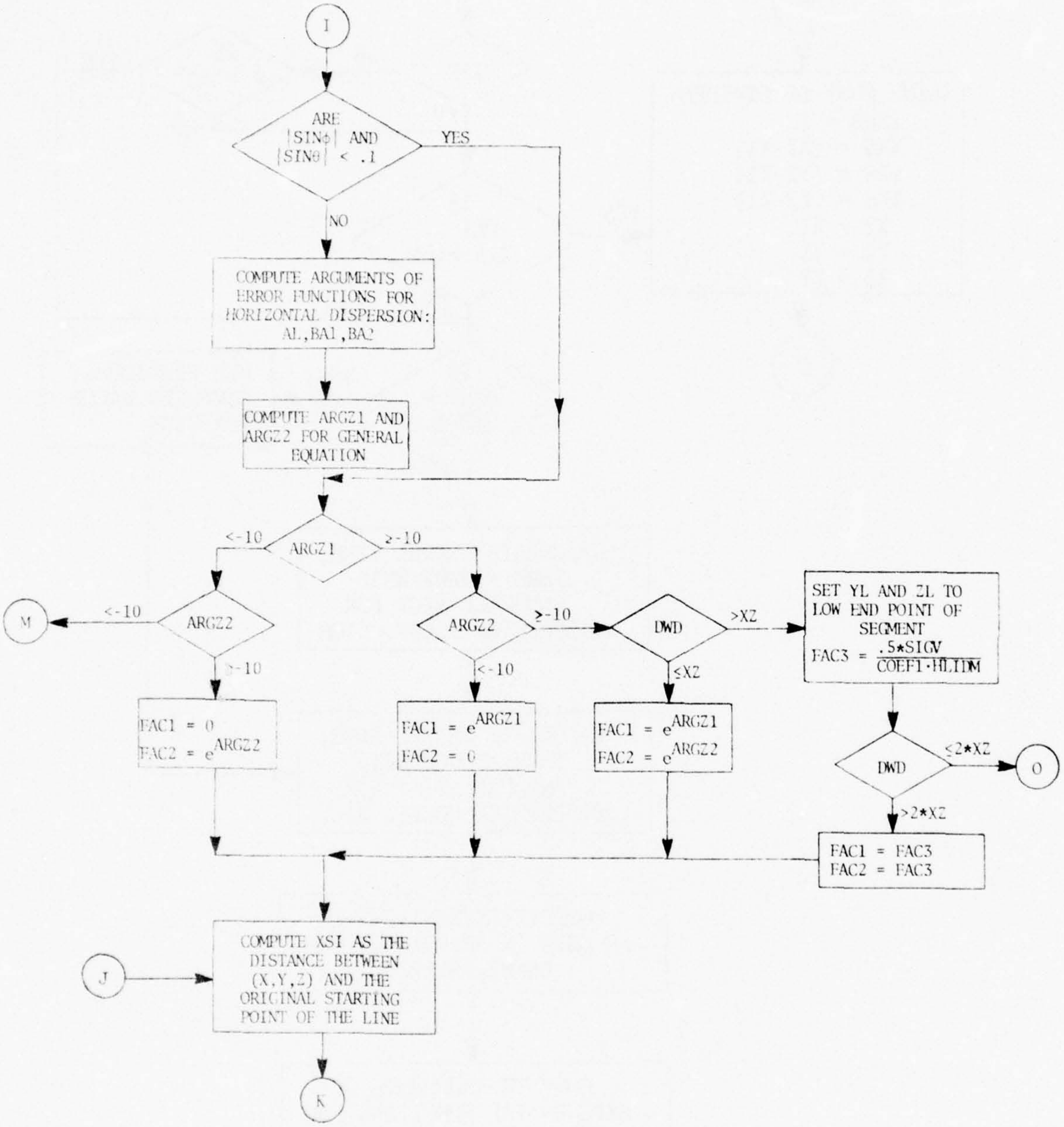

B
ISUB
≠0
=0
DLRS = DL1
COSθ = DL1/DLN
SINθ = (Z2-Z1)DLN
SINφ = (Y2-Y1)/DL1
SET YH AND YL TO THE HIGHER AND LOWER VALUES OF Y1 AND Y2
XMIN = MIN(X1,X2)
IS RECEPTOR UPWIND OF LINE
YES
M
NO
CALCULATE THE DISTANCE BETWEEN RECEPTOR AND UPWIND END OF LINE
XFAR = XR-XMIN
STORE PREVIOUS VALUES OF DWDA AND SIGF
C

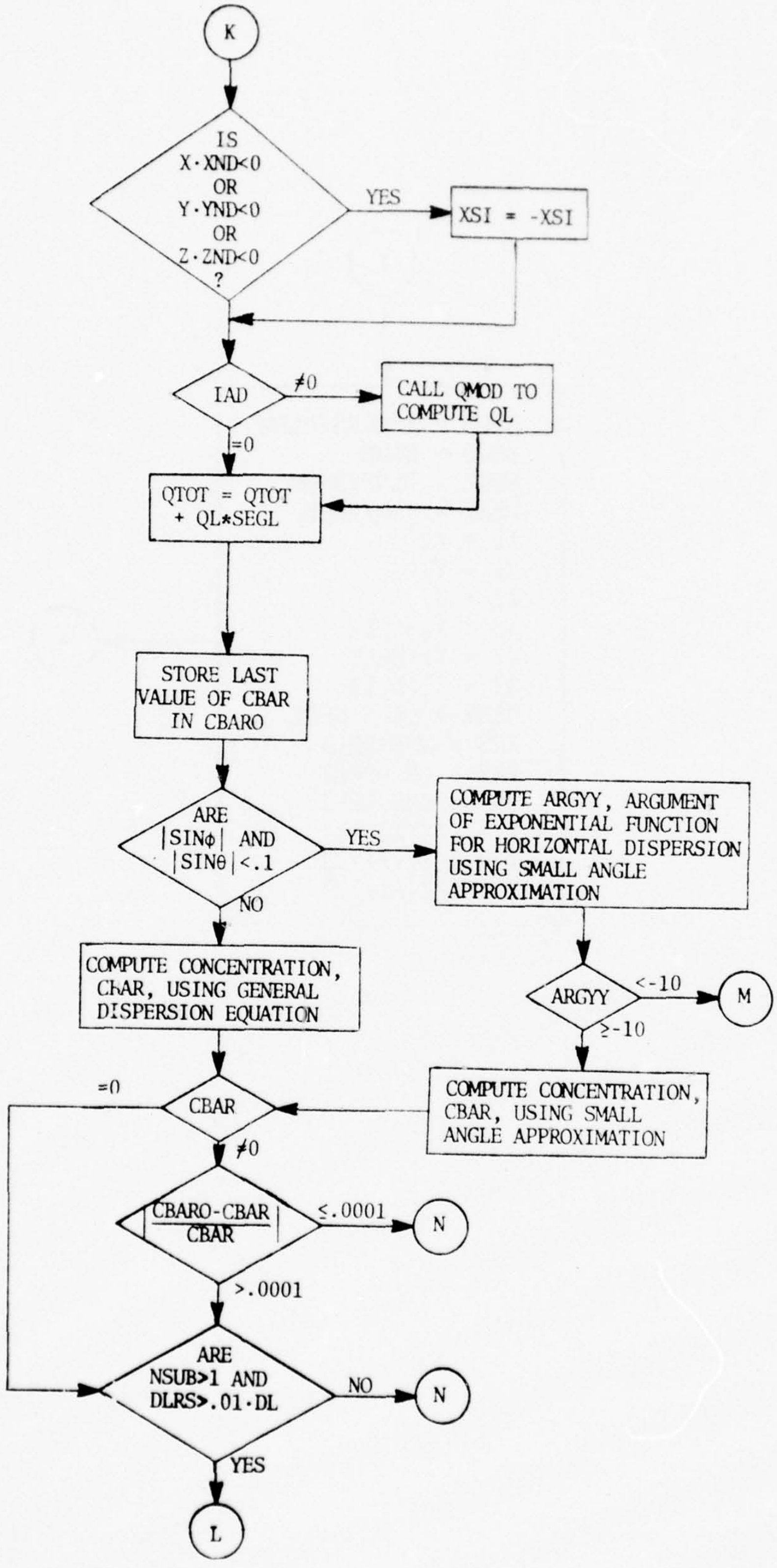

C
CALCULATE DOWNWIND DISTANCES AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE:
DWDA = XFAR+SUDOY
DWDB = XFAR+SUDOZ
TFAR = DWDA/WS36
TFBR = DWDB/WS36
CALL SIGY AND SIGZ TO CALCULATE UPWIND END DISPERSION COEFFICIENTS SIGF AND SIGFZ
STORE COORDINATES OF UPWIND AND DOWNWIND ENDS OF LINE IN XA,YA,ZA AND XB,YB,ZB
ISAC
=1
D
≠1
ARE THE Y AND Z COORDS. OF THE RECEPTOR WITHIN 4 TIMES THE DISPERSION COEFF. OF THE Y AND Z COORDS. OF THE LINE
NO
M
YES
IS THIS A SMALL ANGLE CASE
YES
SET ISAC=1, REDEFINE THE X1,Y1, AND Z1 COORDS. TO CORRESPOND TO X1 = MIN(XR,XA) AND SET (X,Y) AS THE MID-POINT OF THE LINE
A
NO
IAD
≠0
=0
E

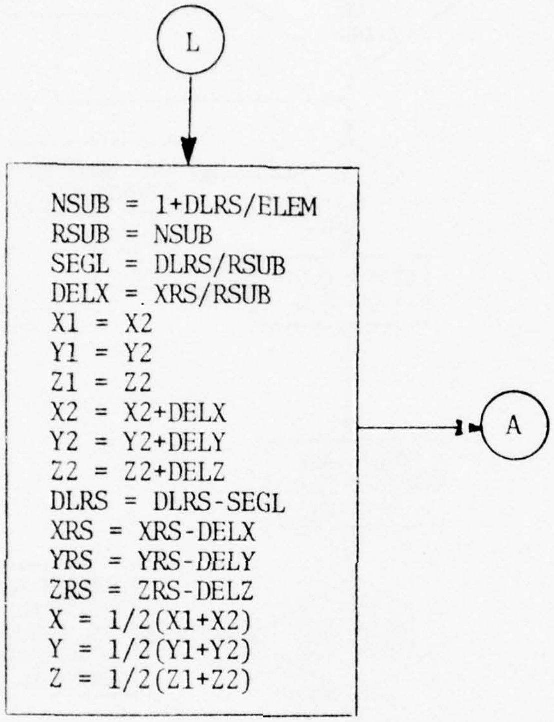

E
ARE THE Y AND Z COORDS. OF THE RECEPTOR WITHIN 3 TIMES THE DISPERSION COEFF. OF THE Y AND Z COORDS. OF THE LINE
NO
M
YES
SET (X,Y) AS THE POINT ON THE LINE WHERE THE CROSS-WIND DISTANCE IS THE LEAST AND THUS IMPACT IS THE GREATEST
D
COMPUTE DOWNWIND DISTANCE DWD = XR-X AND STORE IN DWD1
ISAC
≠1
=1
DWD1 = XR-X1
NSUB
>1
≤1
DWD = DWD1
COMPUTE PSEUDO DOWNWIND DISTANCES
DWDY1 = DWD1+SUDOY
DWDY = DWD+SUDOY
DWDZ = DWD+SUDOZ
F


F
ARE DWDY1,DWDY OR DWDZ ≤0
YES
M
NO
SET THE Z COORD OF THE LINE TO CORRESPOND TO (X,Y)
IS DWDY1 = DWDA or DWDAO
YES
USE PREVIOUSLY COMPUTED VALUE FOR SIGH1
NO
CALCULATE TRAVEL TIME, THRH1 = DWDY1/WS36 AND CALL SIGY FOR DISPERSION COEFF. SIGH1
ELEM = MAX(.2*DWDY1,SIGH1)
IAD
=0
ISAC
≠1
≠0
=1
ISAC = 1
ELEM = .1*ELEM
ELEM = MIN(ELEM,EMIN)
G

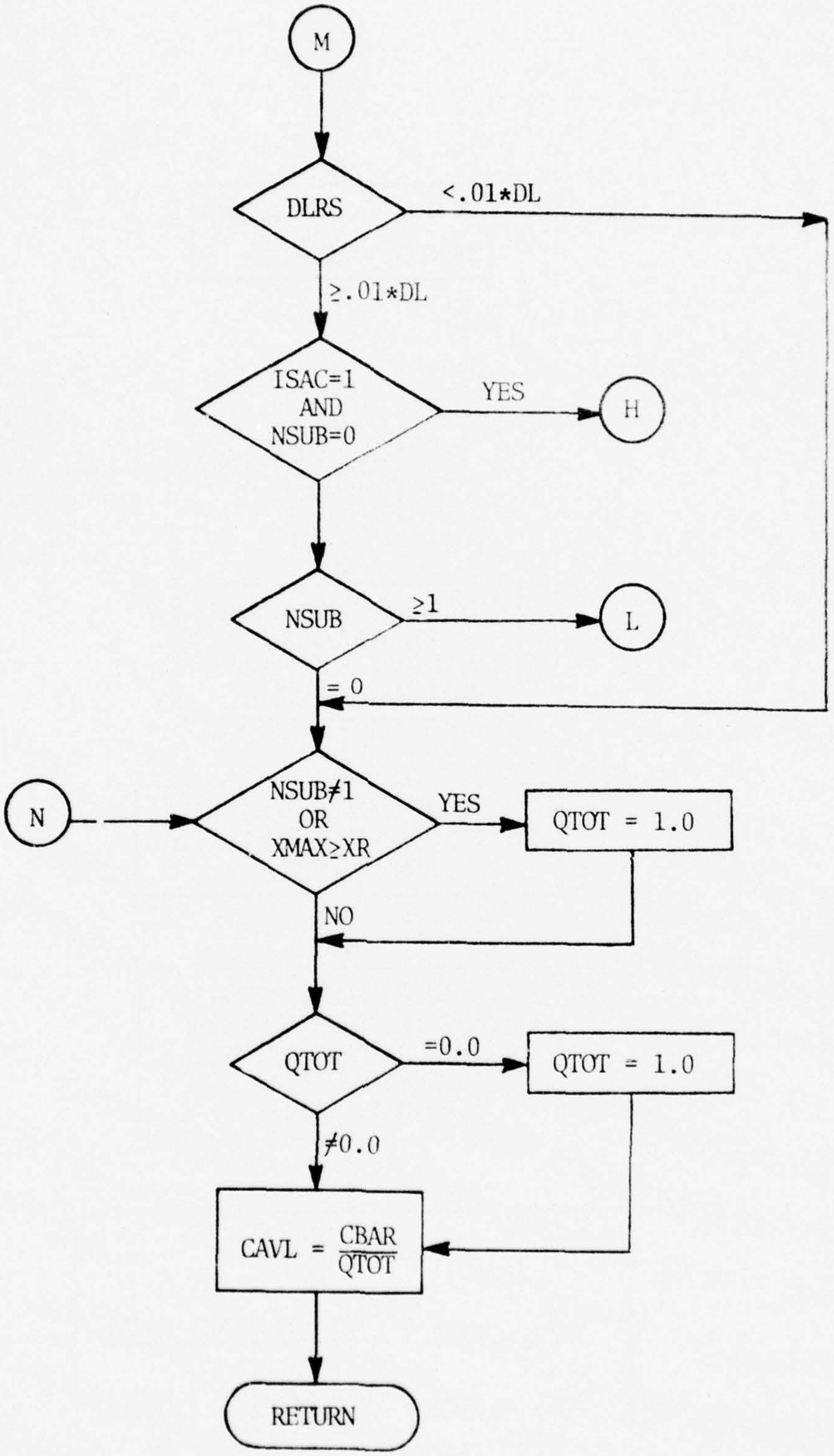

G
H
IAD
=0
≠0
IS
φ > 45°
YES
NO
LINE MUST BE DIVIDED:
ISUB = 1
XRS = (X2-X1)
YRS = (Y2-Y1)
ZRS = (Z2-Z1)
X2 = X1
Y2 = Y1
Z2 = Z1
YES
IS
DL1 >
1.5·ELEM
NO
L
IS
DWDY =
DWDY1
YES
USE PREVIOUSLY
COMPUTED VALUE
FOR SIGH
NO
CALCULATE TRAVEL TIME,
THRH = DWDY/WS36
AND CALL SIGY FOR
DISPERSION COEFF. SIGH
CALCULATE TRAVEL TIME,
THRV = DWDZ/WS36
AND CALL SIGZ FOR
DISPERSION COEFF. SIGV
COMPUTE EXPRESSIONS TO
BE USED IN APPROXIMATIONS:
DENH2,DENZ2,D
COMPUTE ARGUMENTS OF
EXPONENTIAL FUNCTIONS FOR
VERTICAL DISPERSION USING
SMALL ANGLE APPROXIMATION
ARGZ1,ARGZ2
I

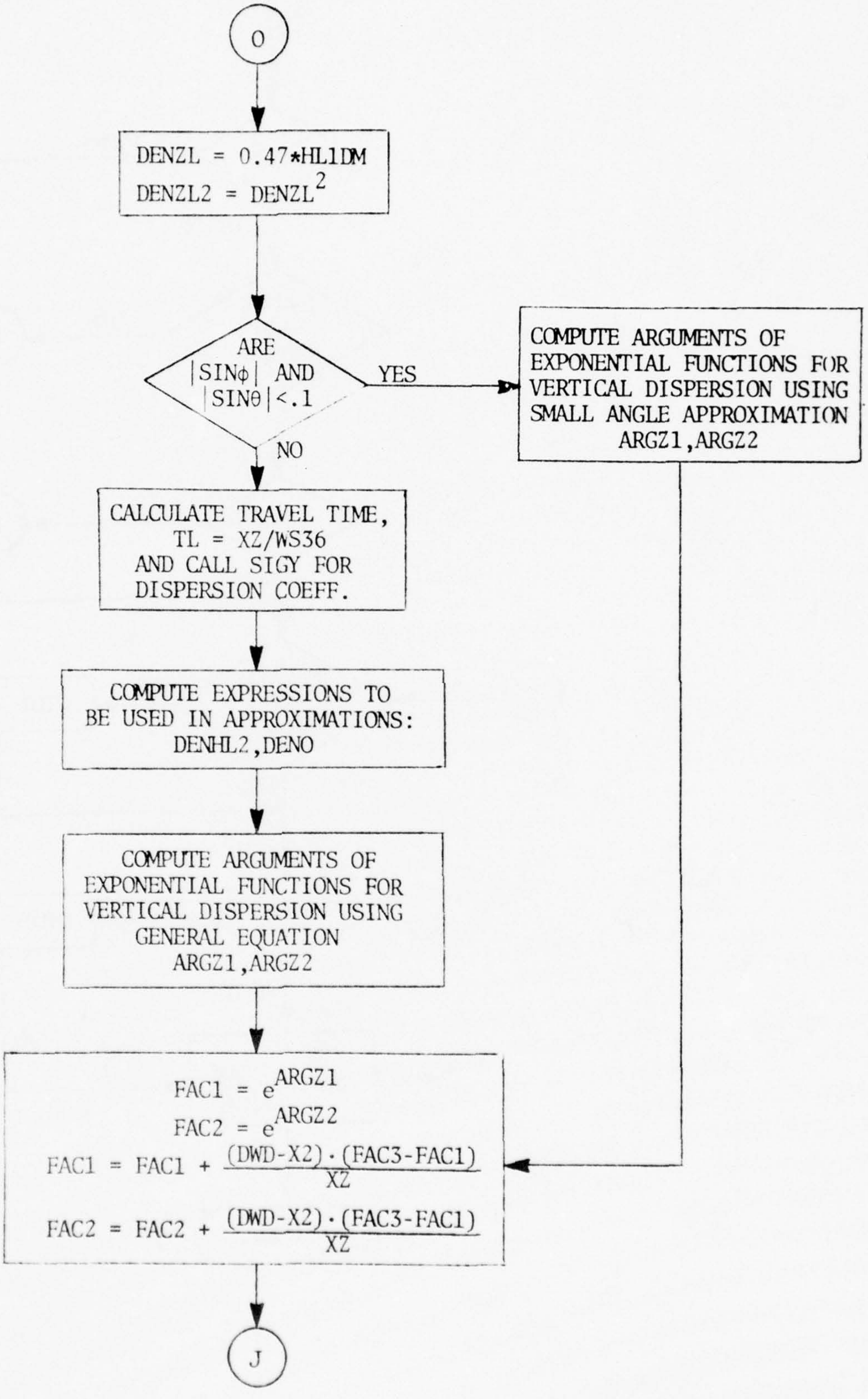

I
ARE
|SINϕ| AND
|SINθ| < .1
YES
NO
COMPUTE ARGUMENTS OF ERROR FUNCTIONS FOR HORIZONTAL DISPERSION: AL,BA1,BA2
COMPUTE ARGZ1 AND ARGZ2 FOR GENERAL EQUATION
<-10
ARGZ1
≥-10
M
<-10
ARGZ2
≥-10
FAC1 = 0
FAC2 = e^ARGZ2
ARGZ2
≥-10
<-10
FAC1 = e^ARGZ1
FAC2 = 0
DWD
>XZ
≤XZ
FAC1 = e^ARGZ1
FAC2 = e^ARGZ2
SET YL AND ZL TO LOW END POINT OF SEGMENT
FAC3 = .5*SIGV / COEF1·HLIDM
DWD
≤2*XZ
O
>2*XZ
FAC1 = FAC3
FAC2 = FAC3
J
COMPUTE XSI AS THE DISTANCE BETWEEN (X,Y,Z) AND THE ORIGINAL STARTING POINT OF THE LINE
K


K
IS
X·XND<0
OR
Y·YND<0
OR
Z·ZND<0
?
YES
XSI = -XSI
IAD
≠0
CALL QMOD TO
COMPUTE QL
=0
QTOT = QTOT
+ QL*SEGL
STORE LAST
VALUE OF CBAR
IN CBARO
ARE
|SINφ| AND
|SINθ|<.1
YES
NO
COMPUTE ARGYY, ARGUMENT
OF EXPONENTIAL FUNCTION
FOR HORIZONTAL DISPERSION
USING SMALL ANGLE
APPROXIMATION
ARGYY
<-10
M
≥-10
COMPUTE CONCENTRATION,
CBAR, USING GENERAL
DISPERSION EQUATION
COMPUTE CONCENTRATION,
CBAR, USING SMALL
ANGLE APPROXIMATION
=0
CBAR
≠0
|CBARO-CBAR / CBAR|
≤.0001
N
>.0001
ARE
NSUB>1 AND
DLRS>.01·DL
NO
N
YES
L



```
NSUB = 1+DLRS/ELEM
RSUB = NSUB
SEGL = DLRS/RSUB
DELX = XRS/RSUB
X1 = X2
Y1 = Y2
Z1 = Z2
X2 = X2+DELX
Y2 = Y2+DELY
Z2 = Z2+DELZ
DLRS = DLRS-SEGL
XRS = XRS-DELX
YRS = YRS-DELY
ZRS = ZRS-DELZ
X = 1/2(X1+X2)
Y = 1/2(Y1+Y2)
Z = 1/2(Z1+Z2)
```


M
DLRS
<.01*DL
≥.01*DL
ISAC=1 AND NSUB=0
YES
H
NSUB
≥1
L
= 0
N
NSUB≠1 OR XMAX≥XR
YES
QTOT = 1.0
NO
QTOT
=0.0
QTOT = 1.0
≠0.0
CAVL = CBAR/QTOT
RETURN


O
DENZL = 0.47*HL1DM
DENZL2 = DENZL^2
ARE |SINφ| AND |SINθ|<.1
YES
NO
COMPUTE ARGUMENTS OF EXPONENTIAL FUNCTIONS FOR VERTICAL DISPERSION USING SMALL ANGLE APPROXIMATION ARGZ1,ARGZ2
CALCULATE TRAVEL TIME, TL = XZ/WS36 AND CALL SIGY FOR DISPERSION COEFF.
COMPUTE EXPRESSIONS TO BE USED IN APPROXIMATIONS: DENHL2,DENO
COMPUTE ARGUMENTS OF EXPONENTIAL FUNCTIONS FOR VERTICAL DISPERSION USING GENERAL EQUATION ARGZ1,ARGZ2
$FAC1 = e^{ARGZ1}$
$FAC2 = e^{ARGZ2}$
$FAC1 = FAC1 + \frac{(DWD-X2)\cdot(FAC3-FAC1)}{XZ}$
$FAC2 = FAC2 + \frac{(DWD-X2)\cdot(FAC3-FAC1)}{XZ}$
J

```
      FUNCTION CAVL(XR,YR,ZR)
C
C   THIS FUNCTION COMPUTES THE POLLUTANT CONCENTRATION DUE TO A
C   FINITE LINE SOURCE
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X5,Y5,Z5,W,DELZ,X6,Y6,Z6,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,B,V12,VS,
     . WS2,WSC,RR,SP,XST,YST,ZST,YND,YND,ZND
      COMMON /XTRAN/ XZ,WSMD,TY,TZ
      DATA COEF1 /.39894/,COEF2 /.31831/
      DATA CAN/0.7071/,EMIN/9.144/
C
C   INITIALIZE COUNTERS, FLAGS AND VARIABLES
C
      ISUB=0
      NSUB=0
      ISAC=0
      LSAC=0
      CBAR=0.
      DWDA=0.
      QTOT=0.
      SEGI=1.0
      WS36=WS*3600.
      HLIDM=HLID
      QL = 1./DL
C
C   INTRODUCE A GENERAL SET OF NOTATION SO THAT THE SAME
C   DISPERSION CALCULATION CAN BE USED FOR THE SMALL ANGLE CASE
C   WHERE THE LINE IS FURTHER SEGMENTED.  X1,Y1,Z1 NOW REFER TO THE
C   LOW END OF THE LINE.
C
      X1=XW1
      Y1=YW1
      Z1=ZW1
      X2=XW2
      Y2=YW2
      Z2=ZW2
      XMAX=AMAX1(X1,X2)
C
C   CALCULATE LENGTH OF LINE
C
    5 DLXY=(X2-X1)**2+(Y2-Y1)**2
      DL1=SQRT(DLXY)
      IF (DL1.EQ.0.AND.Z1.EQ.Z2)GO TO 600
      DLXYZ=DLXY+(Z2-Z1)**2
      DLN=SQRT(DLXYZ)
      IF(ISUB.NE.0) GO TO 6
C
C   THE FIRST TIME THRU, CALCULATE ANGLE OF ELEVATION, THETA,
C   AND ANGLE RELATIVE TO THE X-AXIS, PHI
C
      DLRS=DL1
      CSTH=DL1/DLN
      SNTH=(Z2-Z1)/DLN
      PROJL=Y2-Y1
      IF(ABS(PROJL).LT.1.E-20)PROJL=0.
      SNFI=PROJL/DL1
      ASNF=ABS(SNFI)
C
```

```
C     FIND HIGH AND LOW ENDS OF LINE AS PROJECTED ON THE X-Y PLANE
C
    6 CONTINUE
      IF(Y1.GT.Y2) GO TO 1
      YH=Y2
      YL=Y1
      GO TO 2
    1 YH=Y1
      YL=Y2
    2 CONTINUE
C
C     TEST THE RECEPTOR LOCATION RELATIVE TO THE LINE SOURCE
C
      XMIN=AMIN1(X1,X2)
      IF ((XMIN-XR).GE.0.5) GO TO 500
C
C     RECEPTOR IS DOWNWIND, FIND DISTANCE TO UPWIND END OF LINE
C
      XFAR=XR-XMIN
C
C     STORE PREVIOUS VALUES AND COMPUTE NEW DOWNWIND DISTANCES
C     AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE
C
      DWDAO=DWDA
      SIGFO=SIGF
      DWDA=XFAR+SUDOY
      DWDB=XFAR+SUDOZ
      TFAR=DWDA/WS36
      TFBR=DWDB/WS36
C
C     COMPUTE UPWIND END DISPERSION COEFFICIENTS
C
      SIGF=SIGY(JSTAB,TFAR)
      SIGFZ=SIGZ(JSTAB,TFBR)
C
C     STORE LINE COORDINATES
C
      IF(X1.LE.X2) GO TO 21
      XA=X2
      YA=Y2
      XB=X1
      YB=Y1
      ZB=Z1
      GO TO 22
   21 XA=X1
      YA=Y1
      XB=X2
      YB=Y2
      ZB=Z2
   22 CONTINUE
      IF(ISAC.EQ.1) GO TO 4
C
C     ARE Y AND Z COORDS OF RECEPTOR WITHIN 4 TIMES THE DISPERSION
C     COEFFICIENT OF THE Y AND Z COORDS OF THE LINE
C
      IF(YR.GT.(YH+4.*SIGF)) GO TO 500
      IF(YR.LT.(YL-4.*SIGF)) GO TO 500
      IF(ZR.GT.(Z2+4.*SIGFZ)) GO TO 500
      IF(ZR.LT.(Z1-4.*SIGFZ)) GO TO 500
      IF (ASNF .LT.  CAN .AND. ABS(SNTH) .LT. CAN) GO TO 3
      IF (IAD.NE.0) GO TO 3
C
```

```
C   ANGLE IS LARGE:  ARE THE RECEPTOR COORDS WITHIN 3 TIMES THE
C   DISPERSION COEFFICIENT OF THE LINE COORDS.
C
      IF(YR.GT.(YH+3.*SIGF)) GO TO 500
      IF(YR.LT.(YL-3.*SIGF)) GO TO 500
      IF(ZR.GT.(Z2+3.*SIGFZ)) GO TO 500
      IF(ZR.LT.(Z1-3.*SIGFZ)) GO TO 500
C
C   SET (X,Y) AS POINT ON LINE WHERE IMPACT IS GREATEST
C
      X=X1+(YR-Y1)*(X2-X1)/(Y2-Y1)
      IF(X.GT.XB) GO TO 333
      IF(X.LT.XA) GO TO 33
      Y=YR
      GO TO 4
C
C   ANGLE IS SMALL:  REDEFINE LINE COORDS AND SET (X,Y) AS
C   MIDPOINT OF SEGMENT
C
    3 IF(ASNF.LT.0.1.AND.(ABS(SNTH)).LT.0.1) ISAC=1
      ISAC=1
   30 X=AMIN1(XR,XA)
      Y1=Y1+(X-X1)*(Y2-Y1)/(X2-X1)
      Z1=Z1+(X-X1)*(Z2-Z1)/(X2-X1)
      X1=X
      X2=XB
      Y2=YB
      Z2=ZB
      X=0.5*(X1+X2)
      Y=0.5*(Y1+Y2)
      GO TO 5
   33 X=XA
      Y=YA
      GO TO 4
  333 X=XB
      Y=YB
C
C   COMPUTE DOWNWIND DISTANCE
C
    4 DWD=XR-X
      IF (DWD.LT.-.01) GO TO 30
      DWD1=DWD
      IF (ISAC.NE.1) GO TO 40
      DWD1=XR-X1
      IF (NSUB.LE.1) DWD=DWD1
C
C   COMPUTE PSEUDO DOWNWIND DISTANCES
C
   40 DWDY1=DWD1+SUDOY
      DWDY=DWD+SUDOY
      DWDZ=DWD+SUDOZ
C
C   SET Z COORDINATE OF LINE
C
      IF(X1.EQ.X2) GO TO 44
      Z=Z1+(X-X1)*(Z2-Z1)/(X2-X1)
      GO TO 444
   44 Z=Z1+(Y-Y1)*(Z2-Z1)/(Y2-Y1)
  444 CONTINUE
C
C   COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR
C   PSEUDO DOWNWIND DISTANCE
```

```
C
      IF (DWDY1.EQ.DWDA) GO TO 4111
      IF (DWDY1.EQ.DWDAO) GO TO 4113
      THRH1=DWDY1/WS36
      SIGH1=SIGY(JSTAB,THRH1)
 4211 CONTINUE
C
C   DETERMINE FACTOR TO BE USED IN SUB-DIVIDING THE LINE
C
      ELEM=AMAX1(0.2*DWDY1,SIGH1)
      IF (IAD.NE.0) ISAC=1
      IF(ISAC.EQ.1)ELEM=.1*ELEM
      IF(ELEM.LT.EMIN) ELEM=EMIN
C
C   BRANCH IF ANGLE IS SMALL AND LINE SOURCE IS LONG.
C
      IF (IAD.NE.0) GO TO 4311
      IF (ASNF.GE.CAN) GO TO 4312
 4311 IF (DL1.GT.(1.5*ELEM)) GO TO 55
C
C   COMPUTE TRAVEL TIME AND DISPERION COEFFICIENT FOR PSEUDO
C   VERTICAL DISTANCE
C
 4312 IF (DWDY.EQ.DWDY1) GO TO 4112
      THRH=DWDY/WS36
      SIGH=SIGY(JSTAB,THRH)
 4212 CONTINUE
      THRV=DWDZ/WS36
      SIGV=SIGZ(JSTAB,THRV)
C
C   EXPRESSIONS TO BE USED IN APPROXIMATIONS
C
      DENH2=2.*SIGH**2
      DENZ2=2.*SIGV**2
      D=SIGH*SIGV
C
C   ARGUMENTS OF EXPONENTIAL FUNCTION FOR VERTICAL DISPERSION
C   USING SMALL ANGLE APPROXIMATION
C
      ARGZ1=-(ZR-Z1)**2/DENZ2
      ARGZ2=-(ZR+Z1)**2/DENZ2
      IF (ISAC.EQ.1) GO TO 446
      GO TO 445
 4111 SIGH1=SIGF
      GO TO 4211
 4112 SIGH=SIGH1
      GO TO 4212
 4113 SIGH1=SIGFO
      GO TO 4211
C
C   LARGE ANGLE CASE:  ARGUMENTS OF ERROR FUNCTIONS FOR
C   HORIZONTAL DISPERSION
C
  445 CONTINUE
      ARG=CSTH**2*SNFI**2*SIGV**2+SNTH**2*SIGH**2
      RARG=SQRT(ARG)
      A=RARG/(1.4142*D)
      AL=DLN*A
      ARG1=(YR-Y1)*CSTH*SNFI*SIGV**2
      ARG21=(ZR-Z1)*SNTH*SIGH**2
      ARG22=-(ZR+Z1)*SNTH*SIGH**2
      RA1=-(ARG1+ARG21)/(1.4142*D*RARG)
      BA2=-(ARG1+ARG22)/(1.4142*D*BARG)                                 CAVL0248
      C1=(YR-Y1)**2/DENH2-ARGZ1                                         CAVL0249
      C2=(YR-Y1)**2/DENH2-ARGZ2                                         CAVL0250
C                                                                       CAVL0251
C   ARGUMENTS OF EXPONENTIAL FUNCITONS FOR VERTICAL DISPERSION          CAVL0252
C   USING THE GENERAL EQUATION                                          CAVL0253
C                                                                       CAVL0254
      ARGZ1=BA1**2-C1                                                   CAVL0255
      ARGZ2=BA2**2-C2                                                   CAVL0256
  446 IF(ARGZ1.LT.-10.)GOTO 2411                                        CAVL0257
      IF(ARGZ2.GE.-10.)GOTO 2412                                        CAVL0258
      FAC1=EXP(ARGZ1)                                                   CAVL0259
      FAC2=0                                                            CAVL0260
      GO TO 39                                                          CAVL0261
 2411 IF (ARGZ2.LT.-10) GO TO 500                                       CAVL0262
      FAC1=0                                                            CAVL0263
      FAC2=EXP(ARGZ2)                                                   CAVL0264
      GO TO 39                                                          CAVL0265
 2412 IF(DWD.GT.XZ) GO TO 100                                           CAVL0266
C                                                                       CAVL0267
C   DOWNWIND DISTANCE IS LESS THAN THE CRITICAL DISTANCE:  ONLY         CAVL0268
C   SOURCE AND GROUND REFLECTION ARE CONSIDERED                         CAVL0269
C                                                                       CAVL0270
      FAC1=EXP(ARGZ1)                                                   CAVL0271
      FAC2=EXP(ARGZ2)                                                   CAVL0272
   39 CONTINUE                                                          CAVL0273
C                                                                       CAVL0274
C   FIND THE LINEAR DISTRIBUTION OF POLLUTION ON THE                    CAVL0275
C   RUNWAY FOR LANDING AND TAKE-OFF                                     CAVL0276
C                                                                       CAVL0277
      XSI2=(X-XST)**2+(Y-YST)**2+(Z-ZST)**2                             CAVL0278
      XSI=SQRT(XSI2)                                                    CAVL0279
      IF(X*XND.LT.0.OR.Y*YND.LT.0.OR.Z*ZND.LT.0) XSI=-XSI               CAVL0280
      IF (IAD .NE. 0) CALL QMOD(XSI,QL)                                 CAVL0281
      QTOT=QTOT+QL*SEGL                                                 CAVL0282
C                                                                       CAVL0283
C   STORE LAST VALUE OF CBAR                                            CAVL0284
C                                                                       CAVL0285
      CBARO=CBAR                                                        CAVL0286
      IF(ISAC.EQ.1) GO TO 50                                            CAVL0287
C                                                                       CAVL0288
C   GENERAL DISPERSION EQUATION                                         CAVL0289
C                                                                       CAVL0290
      FJ1=FAC1*DIFERF(BA1,AL)                                           CAVL0291
      FJ2=FAC2*DIFERF(BA2,AL)                                           CAVL0292
      CBAR=CBAR+0.35355*COEF1*QI*(FJ1+FJ2)/(A*D)                        CAVL0293
  499 IF (CBARO.EQ.0) GO TO 49                                          CAVL0294
      IF(ABS((CBARO-CBAR)/CBAR).LE..00010) GO TO 600                    CAVL0295
   49 CONTINUE                                                          CAVL0296
      IF(NSUB.GT.1.AND.DLPS.GT.(.01*DL)) GO TO 60                       CAVL0297
      GO TO 600                                                         CAVL0298
C                                                                       CAVL0299
C   SMALL-ANGLE APPROXIMATION                                           CAVL0300
C                                                                       CAVL0301
   50 ARGYY=-(YR-Y1)**2/DENH2                                           CAVL0302
      IF(ARGYY.LT.-10.) GO TO 500                                       CAVL0303
      FAC=0.5*(FAC1+FAC2)                                               CAVL0304
      BRAC=EXP(ARGYY)                                                   CAVL0305
      CBAR=CBAR+COEF2*QL*DLN*FAC*BRAC/D                                 CAVL0306
      GO TO 499                                                         CAVL0307
C                                                                       CAVL0308
C   ANGLE IS SMALL AND SOURCE IS LONG                                   CAVL0309
```

```
C
   55 ISUB=1
      XRS=X2-X1
      YRS=Y2-Y1
      ZRS=Z2-Z1
      X2=X1
      Y2=Y1
      Z2=Z1
C
C    COMPUTE COORDINATES FOR NEXT LINE SEGMENT
C
   60 NSUB=1.+DLRS/FLEM
      RSUB=NSUB
      SEGL=DLRS/RSUB
      DELX=XRS/RSUB
      DELY=YRS/RSUB
      DELZ=ZRS/RSUB
      X1=X2
      Y1=Y2
      Z1=Z2
      X2=X2+DELY
      Y2=Y2+DELY
      Z2=Z2+DELZ
      DLRS=DLRS-SEGL
      XRS=XRS-DELX
      YRS=YRS-DELY
      ZRS=ZRS-DELZ
      X=.5*(X1+X2)
      Y=.5*(Y1+Y2)
      Z=.5*(Z1+Z2)
C
C    GO BACK TO COMPUTE CONTRIBUTION FROM NEXT SEGMENT
C
      GO TO 5
C
C    DOWNWIND DISTANCE IS GREATER THAN, BUT LESS THAN TWICE, THE
C    CRITICAL DISTANCE.  LINEAR INTERPOLATION IS USED
C
  100  YL = Y1
       ZL = Z1
       IF (Z1 .LE. Z2) GO TO 105
       YL = Y2
       ZL = Z2
  105  FAC3=0.5*SIGV/(COEF1*HLIDM)
       IF(LVL.GT.2.*XZ) GO TO 200
       DENZL=0.47*HLIDM
       DENZL2=DENZL**2
       IF (LSAC.EQ.1) GO TO 101
  102 TL=XZ/WS36
      DENHL2=2.*SIGY(JSTAB,TL)**2
      DENO=CSTH**2*SNFI**2*DENZL2   +SNTH**2*DENHL2
      ARGZ1=-((YR-YL)*SNTH-(ZR-ZL)*CSTH*SNFI)**2/DENO
      ARGZ2=-((YR-YL)*SNTH-(ZR+ZL)*CSTH*SNFI)**2/DENO
      GO TO 103
  101 ARGZ1=-(ZR-ZL)**2/DENZL2
      ARGZ2=-(ZR+ZL)**2/DENZL2
  103 FAC1=EXP(ARGZ1)
      FAC2=EXP(ARGZ2)
      FAC1=FAC1+(DWD-XZ)*(FAC3-FAC1)/XZ
      FAC2=FAC2+(DWD-XZ)*(FAC3-FAC2)/XZ
      GO TO 39
C
```

```
C     DOWNWIND DISTANCE IS BEYOND 2 TIMES THE CRITICAL DISTANCE,        CAVL0372
C     UNIFORM MIXING IS ASSUMED                                         CAVL0373
C                                                                       CAVL0374
  200 FAC1=FAC3                                                         CAVL0375
      FAC2=FAC3                                                         CAVL0376
      GO TO 39                                                          CAVL0377
  500 IF(DLRS.LT.(.01*DL)) GO TO 600                                    CAVL0378
      IF(ISAC.EQ.1.AND.NSUB.EQ.0) GO TO 55                              CAVL0379
      IF(NSUB.GE.1) GO TO 60                                            CAVL0380
  600 IF (NSUB.NE.1.OR.XMAX.GE.XF) QTOT=1.0                             CAVL0381
      IF (QTOT.EQ.0.0) QTOT=1.0                                         CAVL0382
C                                                                       CAVL0383
C     NORMALIZE CBAR TO THE TOTAL POLLUTANT DENSITY CALCULATED          CAVL0384
C     ALONG THE LINE                                                    CAVL0385
C                                                                       CAVL0386
      CAVL=CBAR/QTOT                                                    CAVL0387
      RETURN                                                            CAVL0388
      END                                                               CAVL0389
```

SUBROUTINE CLASSE

Purpose:

To print an error message if the wrong ICLASS value is input to one of the airbase non-aircraft or environ emission distribution subroutines.

Input:

None

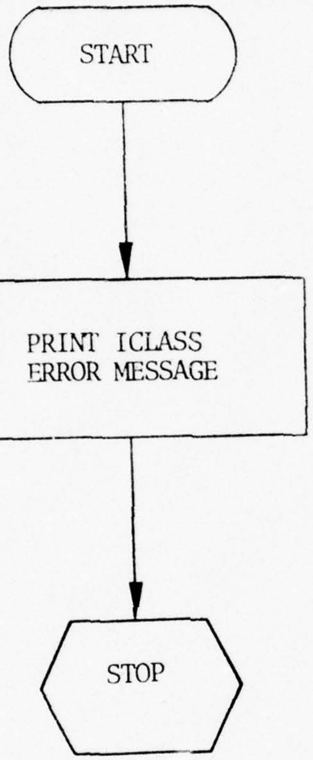
Output:

A message indicating the value of ICLASS set by the code and the value supplied by the user.

Subroutines Called:

None

SUBROUTINE CLASSE



```
      SUBROUTINE CLASSE  (I,J)                                          CLASE000
C                                                                       CLASE001
C     THIS ROUTINE PRINTS THE ICLASS ERROR MESSAGE                      CLASE002
C                                                                       CLASE003
      PRINT 1, I,J                                                      CLASE004
    1 FORMAT(17H0ICLASS SHOULD BE,I4,18H, INPUT CARD READS,I4)          CLASE005
      STOP                                                              CLASE006
      END                                                               CLASE007
```

SUBROUTINE DEPART

Purpose:

To calculate the points in the runway roll and climbout modes as a function of aircraft type using current meteorological conditions and airbase specific pressure altitudes and airbase dependent basic aircraft parameters.

Input:

Basic aircraft data, current meteorological conditions, runway data, aircraft identification.

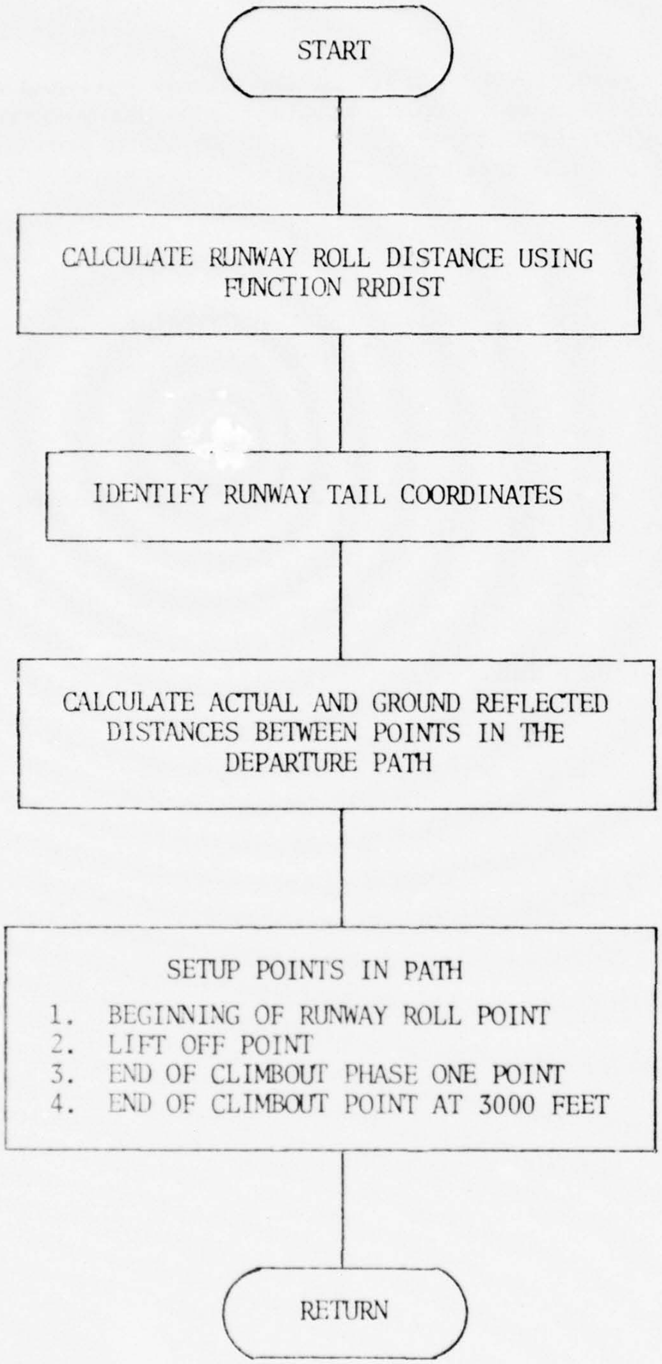
Output:

Points in departure path as a function of runway and aircraft type.

Subroutine Called:

Function RRDIST.

SUBROUTINE DEPART


START
CALCULATE RUNWAY ROLL DISTANCE USING FUNCTION RRDIST
IDENTIFY RUNWAY TAIL COORDINATES
CALCULATE ACTUAL AND GROUND REFLECTED DISTANCES BETWEEN POINTS IN THE DEPARTURE PATH
SETUP POINTS IN PATH
1. BEGINNING OF RUNWAY ROLL POINT
2. LIFT OFF POINT
3. END OF CLIMBOUT PHASE ONE POINT
4. END OF CLIMBOUT POINT AT 3000 FEET
RETURN

```
      SUBROUTINE DEPART(N,I)
C
C     THIS ROUTINE CALCULATES THE POINTS IN THE DEPARTURE PATH
C     AS A FUNCTION OF RUNWAY(N) AND AIRCRAFT TYPE(I)
C
      REAL INDSPD
      INTEGER  ENGNO
      COMMON / MET / WS,WSMPH,IWS,WD,IWD,SINEWD,CVSEWD,JSTAB,HLID,TEMF,
     . TEMK
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DIBAR
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . INDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     .ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JFS1(6)
      RD=RNWY(7,N)
      WSPD=WS*1.9426*COS(WD-RD)
      HDIS12=PRDIST(IDRR(I),PA,TEMF,TOWT(I),WSPD)*3.048E-4
      XA=SIN(RNWY(7,N))
      YA=COS(RNWY(7,N))
      X=RNWY(2,N)
      Y=RNWY(3,N)
      Z=RNWY(4,N)/1000.
      DIS23=COHT1(I)/SIN(ASCNT1(I))
      DIS34=(CLMBHT-COHT1(I))/SIN(ASCNT2(I))
      HDIS23=COHT1(I)/TAN(ASCNT1(I))
      HDIS34=(CLMBHT-COHT1(I))/TAN(ASCNT2(I))
C
C     START OF RUNWAY ROLL DATA
C
      DEPFCN(1,I,N)=X
      DEPFCN(2,I,N)=Y
      DEPFCN(3,I,N)=Z*1000.
      DEPFCN(4,I,N)=0.0
      DEPFCN(5,I,N)=HDIS12
      DEPFCN(6,I,N)=2.0*HDIS12/TOSPD(I)
C
C     POINT OF LIFTOFF DATA
C
      DEPFCN(7,I,N)=X+HDIS12*XA
      DEPFCN(8,I,N)=Y+HDIS12*YA
      DEPFCN(9,I,N)=Z*1000.
      DEPFCN(10,I,N)=TOSPD(I)
      DEPFCN(11,I,N)=DIS23
      DEPFCN(12,I,N)=2.0*DIS23/(TOSPD(I)+COSPD1(I))
C
C     END OF CLIMB1 POINT DATA
C
      DEPFCN(13,I,N)=DEPFCN(7,I,N)+HDIS23*XA
      DEPFCN(14,I,N)=DEPFCN(8,I,N)+HDIS23*YA
      DEPFCN(15,I,N)=COHT1(I)*1000.
      DEPFCN(16,I,N)=COSPD1(I)
      DEPFCN(17,I,N)=DIS34
      DEPFCN(18,I,N)=2.0*DIS34/(COSPD1(I)+COSPD2(I))
C
C     END OF CLIMBOUT POINT DATA
```

```
C                                                                        DEPRT062
      DEPFCN(19,I,N)=DEPFCN(13,I,N)+HDIS34*XA                            DEPRT063
      DEPFCN(20,I,N)=DEPFCN(14,I,N)+HDIS34*YA                            DEPRT064
      DEPFCN(21,I,N)=CLMBHT*1000.                                        DEPRT065
      DEPFCN(22,I,N)=COSPD2(I)                                           DEPRT066
      RETURN                                                             DEPRT067
      END                                                                DEPRT068
```

FUNCTION DIFERF(X,PH)

Purpose:

To find the difference between two error functions, erf(X+PH) - erf(X).

Input:

X and PH

Output:

The difference between the error functions

Procedure:

1. If PH $\leq$ .05, the formula given in the Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 is used:

   $$\mathrm{DIFERF} = 1.12838 \cdot \mathrm{PH} \cdot e^{-X^2} [1 - \mathrm{PH} \cdot X + (2 \cdot X^2 - 1) \cdot \mathrm{PH}^{2/3}]$$

2. If PH > .05 and X and X+PH are of different sign:

   DIFERF = erf(X+PH) - erf(X)

3. If PH > .05 and X and X+PH are both negative:

   DIFERF = -1.*[erfc(-X) - erfc(-X-PH)]

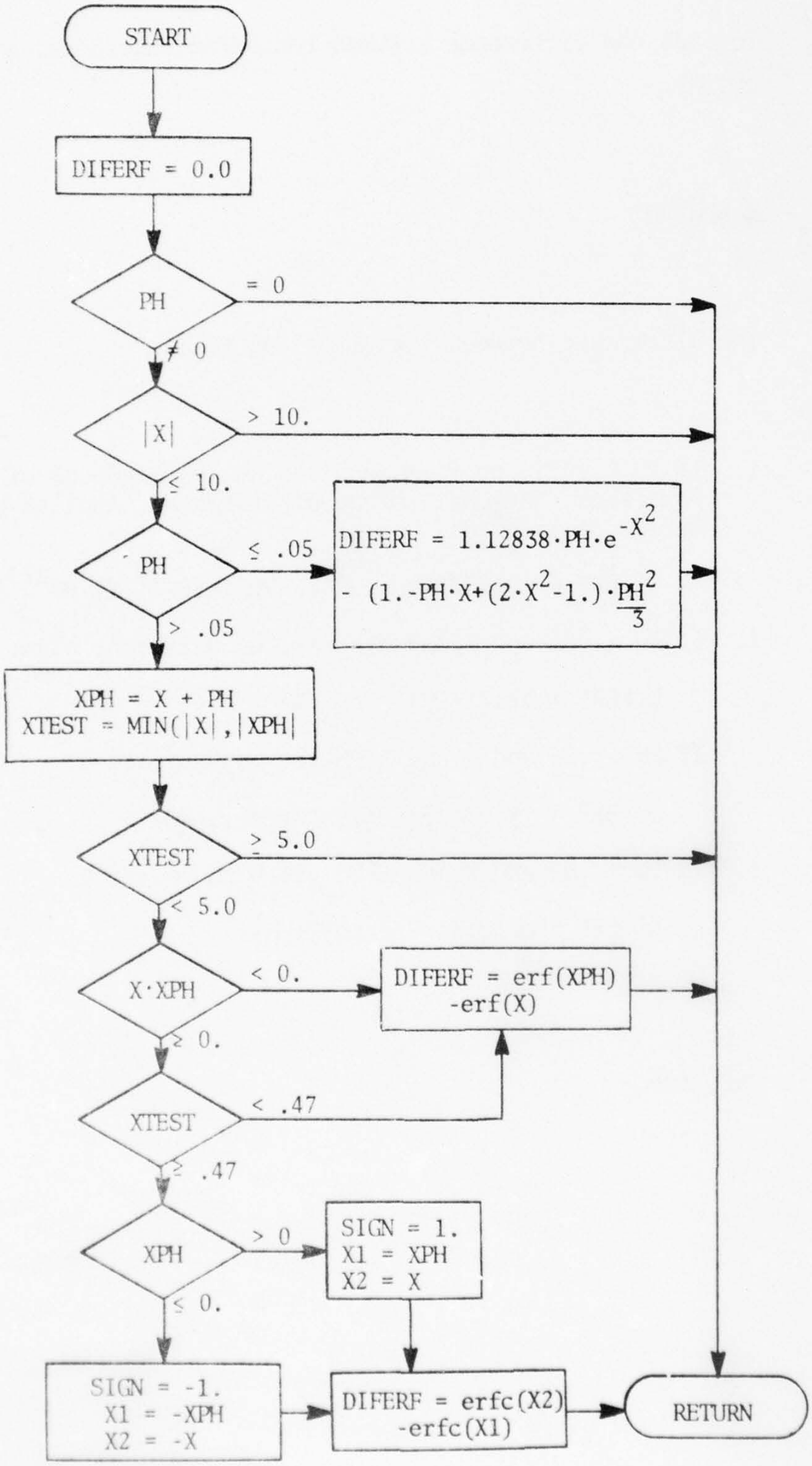
4. If PH > .05 and X and X+PH are both positive:

   DIFERF = erfc(X) - erfc(X+PH)

Function Called:

ERF, ERFC

FUNCTION DIFERF(X,PH)



```
      FUNCTION DIFERF(X,PH)
C
C   THIS FUNCTION FINDS THE DIFFERENCE BETWEEN TWO ERROR FUNCTIONS
C   USING VARYING METHODS BASED ON THE SIZE OF THE ARGUMENTS
C
      DIFERF=0.
      IF (PH.EQ.0.0) GO TO 50
      IF (ABS(X).GT.10.0) GO TO 50
      IF (PH.GT.0.05) GO TO 10
C
C   USE METHOD OUTLINED IN HANDBOOK OF MATH FUNCTIONS, NATL
C   BUREAU OF STANDARDS, APPLIED MATH SERIES 55
C
      DIFERF=(1.12838*PH/EXP(X**2))*(1.-PH*X+(2.*X**2-1.)*PH**2/3.)
      GO TO 50
C
C   DIFFERENCE IS TOO LARGE, MUST USE ERF OR ERFC
C
   10 XPH=X+PH
      XTEST=AMIN1(ABS(X),ABS(X+PH))
      IF (XTEST.GE.5.0) GO TO 50
      IF (X*XPH.LT.0.0) GO TO 40
      IF (XTEST.LT.0.47) GO TO 40
C
C   CAN ONLY REACH HERE WHEN X AND XPH HAVE SAME SIGN
C
      IF (XPH.GT.0.0) GO TO 20
      SIGN=-1.
      X1=-XPH
      X2=-X
      GO TO 30
   20 SIGN=1.
      X1=XPH
      X2=X
   30 DIFERF=SIGN*(ERFC(X2)-ERFC(X1))
      GO TO 50
   40 DIFEFF=ERF(XPH)-ERF(X)
   50 RETURN
      END
```

SUBROUTINE DIFMOD

Purpose:

To loop through all wind directions and speeds, determine the wind dependent sources, and direct the calls to the proper diffusion routine for all input sources.

Input:

1. Restart data
2. Special wind case data
3. Point source data for:
   a. Environs
   b. Airbase
   c. Aircraft
4. Area source data for:
   a. Environs
   b. Airbase
   c. Aircraft
5. Line source data for:
   a. Environ
   b. Airbase
   c. Aircraft

Output:

SORC, a vector which contains data for the current source to be transferred to the diffusion models.

Procedure:

1. Set the receptor data array to zero.
2. Check for restart.
3. For all wind directions and speeds:
   a. Check for special wind cases.
   b. Determine wind dependent sources.
   c. Set the type flag for environs, airbase or aircraft.

d. Fill the SORC vector with the source description and emission parameters.

e. Check for non-zero emissions from this source and call POLUT for point and area sources and PSEUDL and POLUTL for line sources.

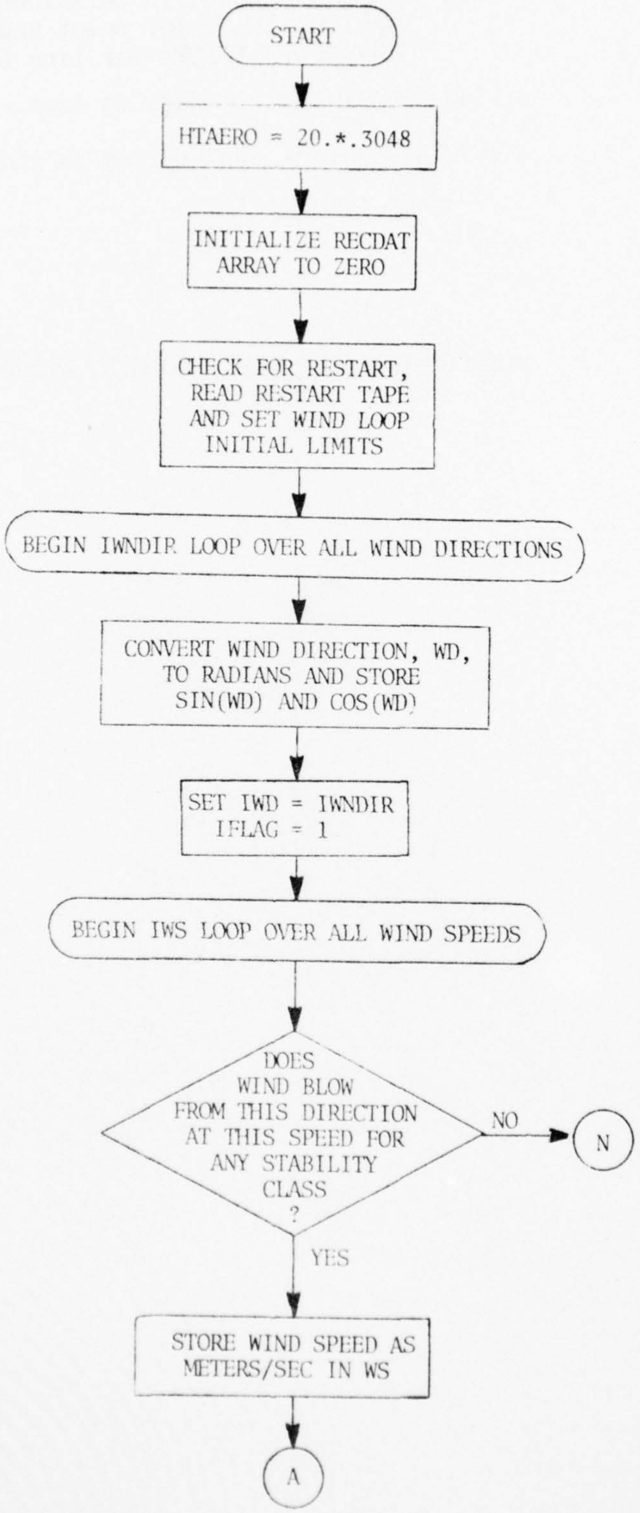
f. Write restart data on tape.

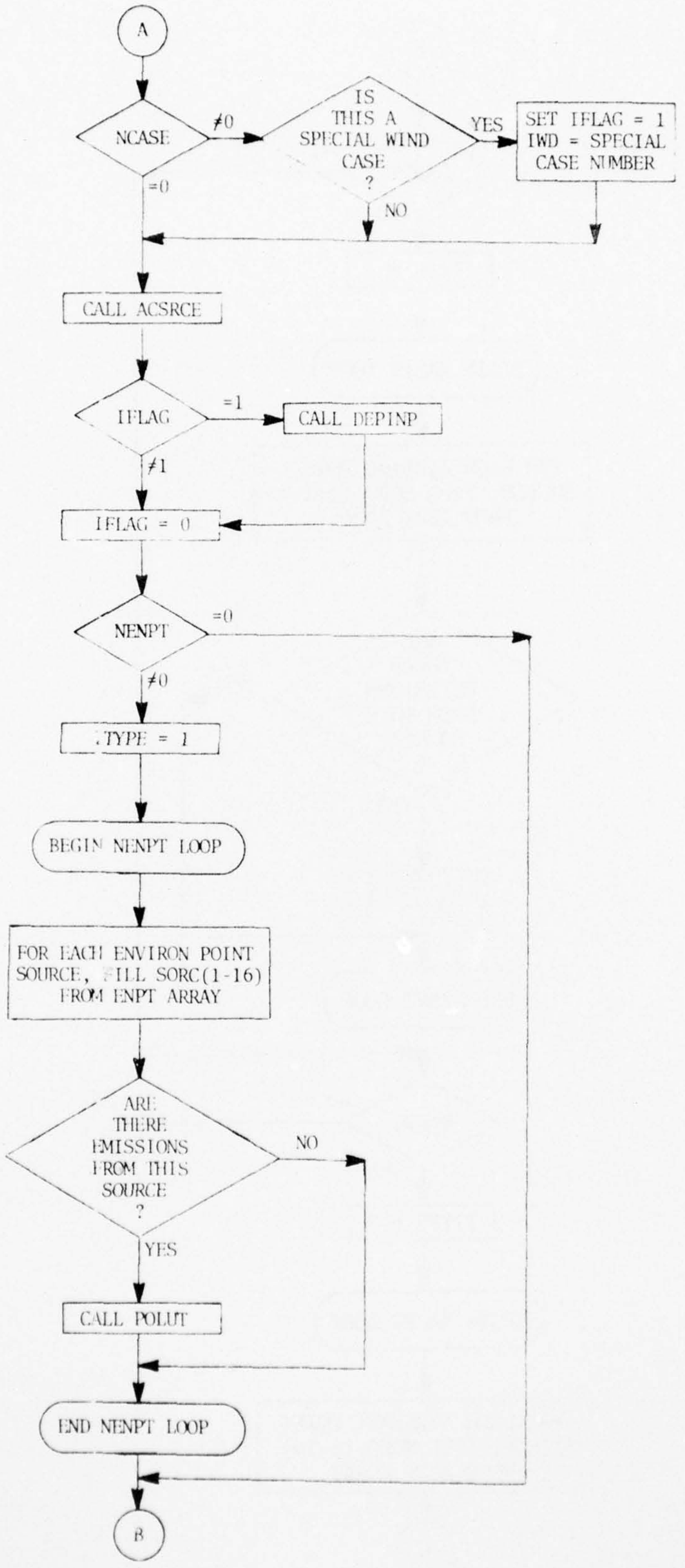
4. If the statistical option is chosen, write a final record on that tape.

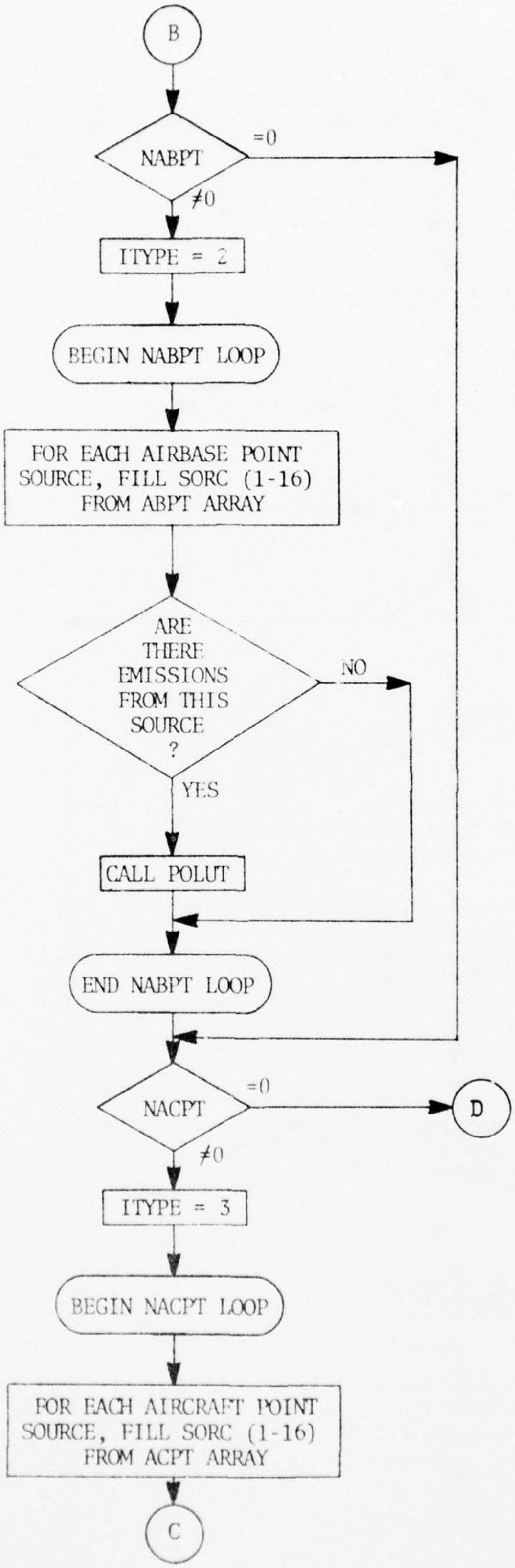
Subroutines Called:

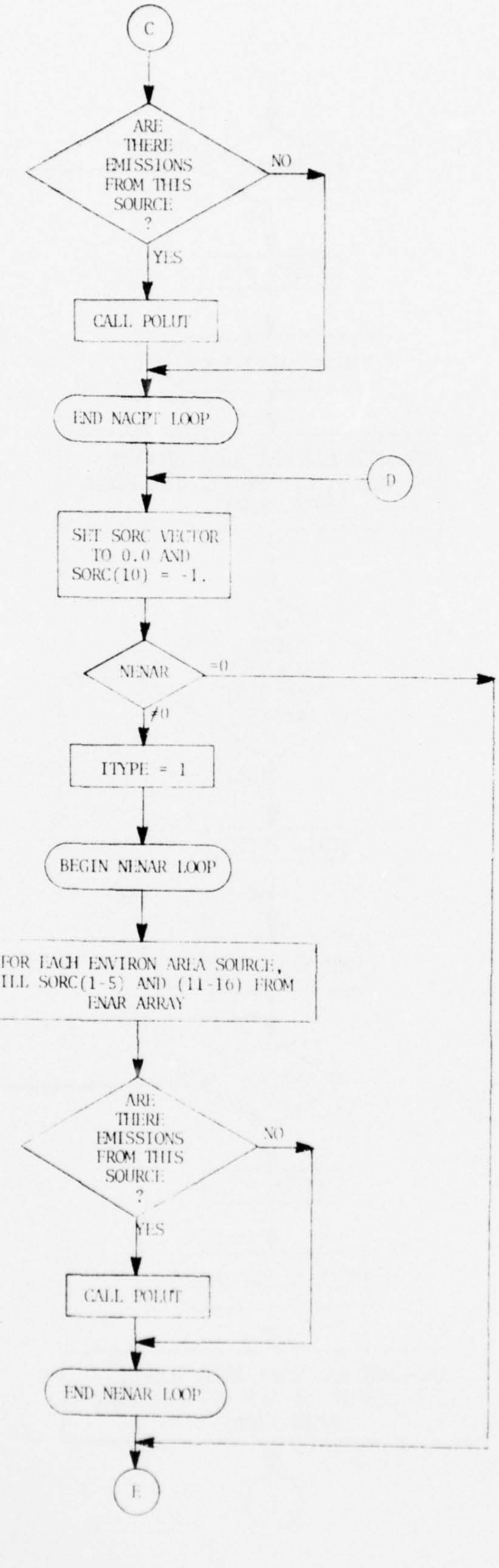
ACSRCE, DEPINP, POLUT, POLUTL, PSEUDL

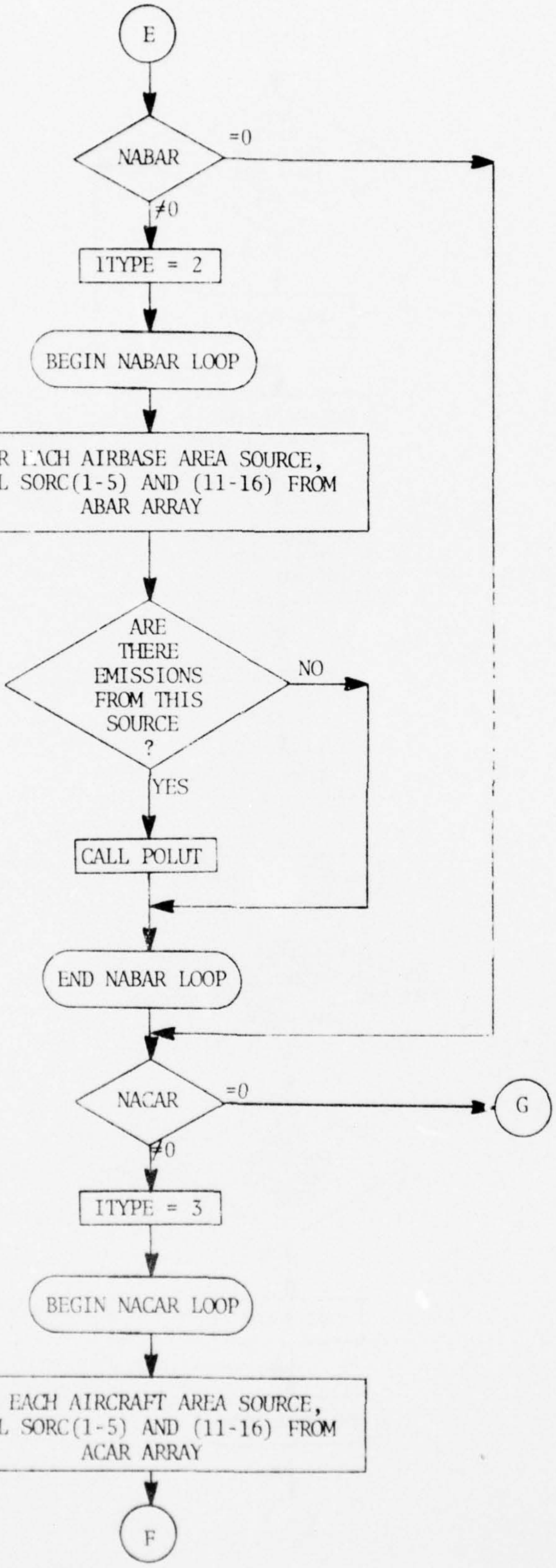
SUBROUTINE DIFMOD

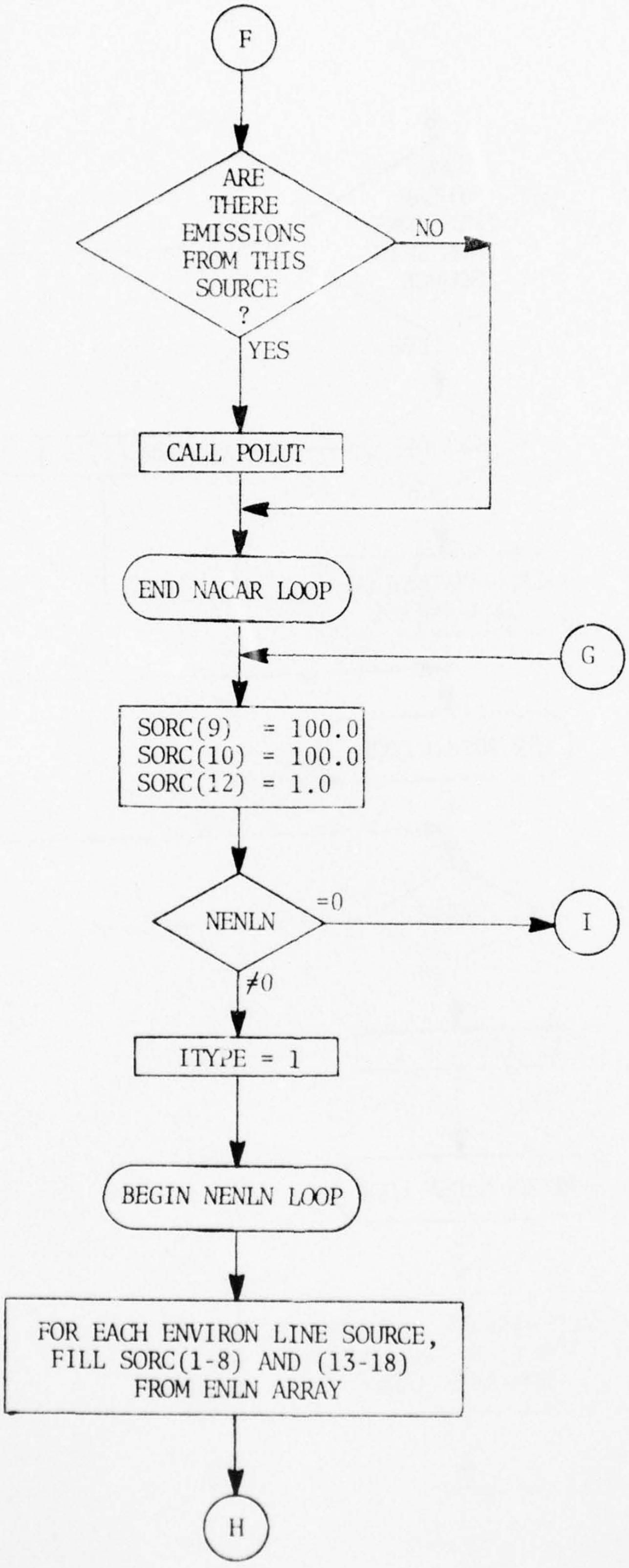

START
HTAERO = 20.*.3048
INITIALIZE RECDAT ARRAY TO ZERO
CHECK FOR RESTART, READ RESTART TAPE AND SET WIND LOOP INITIAL LIMITS
BEGIN IWNDIR LOOP OVER ALL WIND DIRECTIONS
CONVERT WIND DIRECTION, WD, TO RADIANS AND STORE SIN(WD) AND COS(WD)
SET IWD = IWNDIR
IFLAG = 1
BEGIN IWS LOOP OVER ALL WIND SPEEDS
DOES WIND BLOW FROM THIS DIRECTION AT THIS SPEED FOR ANY STABILITY CLASS ?
NO
N
YES
STORE WIND SPEED AS METERS/SEC IN WS
A

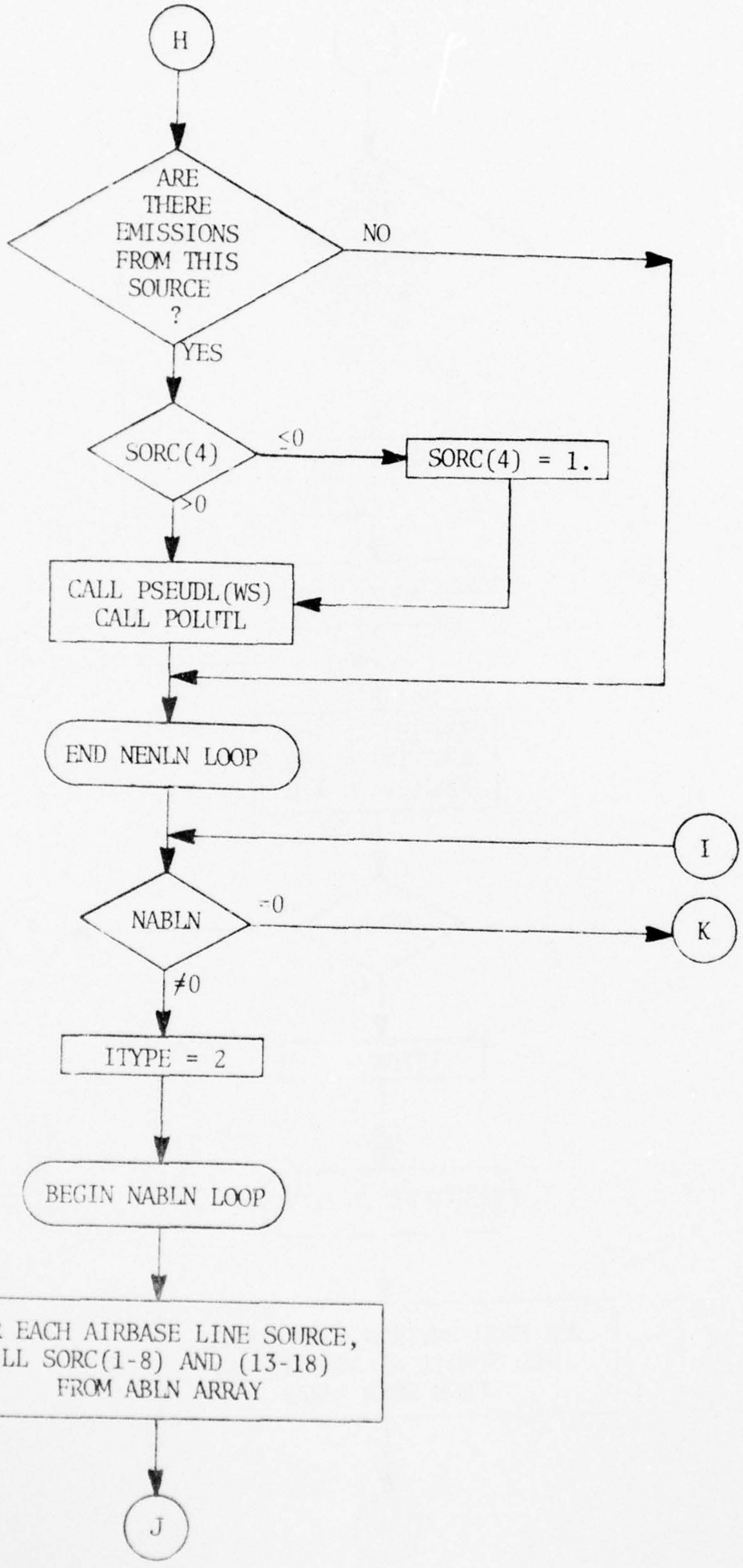

A
NCASE
≠0
IS THIS A SPECIAL WIND CASE ?
YES
SET IFLAG = 1
IWD = SPECIAL CASE NUMBER
NO
=0
CALL ACSRCE
IFLAG
=1
CALL DEPINP
≠1
IFLAG = 0
NENPT
=0
≠0
ITYPE = 1
BEGIN NENPT LOOP
FOR EACH ENVIRON POINT SOURCE, FILL SORC(1-16) FROM ENPT ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NENPT LOOP
B

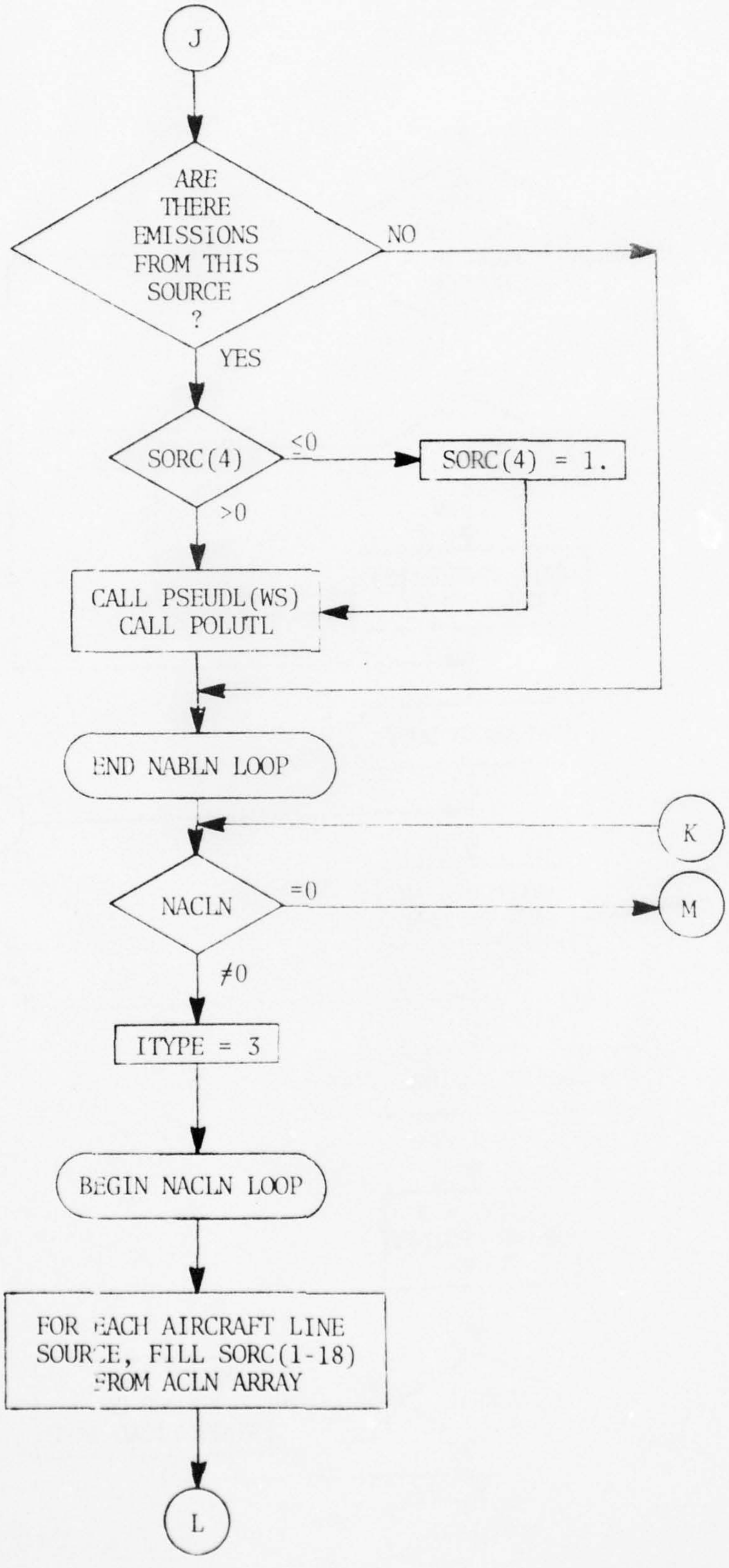

B
NABPT
=0
≠0
ITYPE = 2
BEGIN NABPT LOOP
FOR EACH AIRBASE POINT SOURCE, FILL SORC (1-16) FROM ABPT ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NABPT LOOP
NACPT
=0
D
≠0
ITYPE = 3
BEGIN NACPT LOOP
FOR EACH AIRCRAFT POINT SOURCE, FILL SORC (1-16) FROM ACPT ARRAY
C

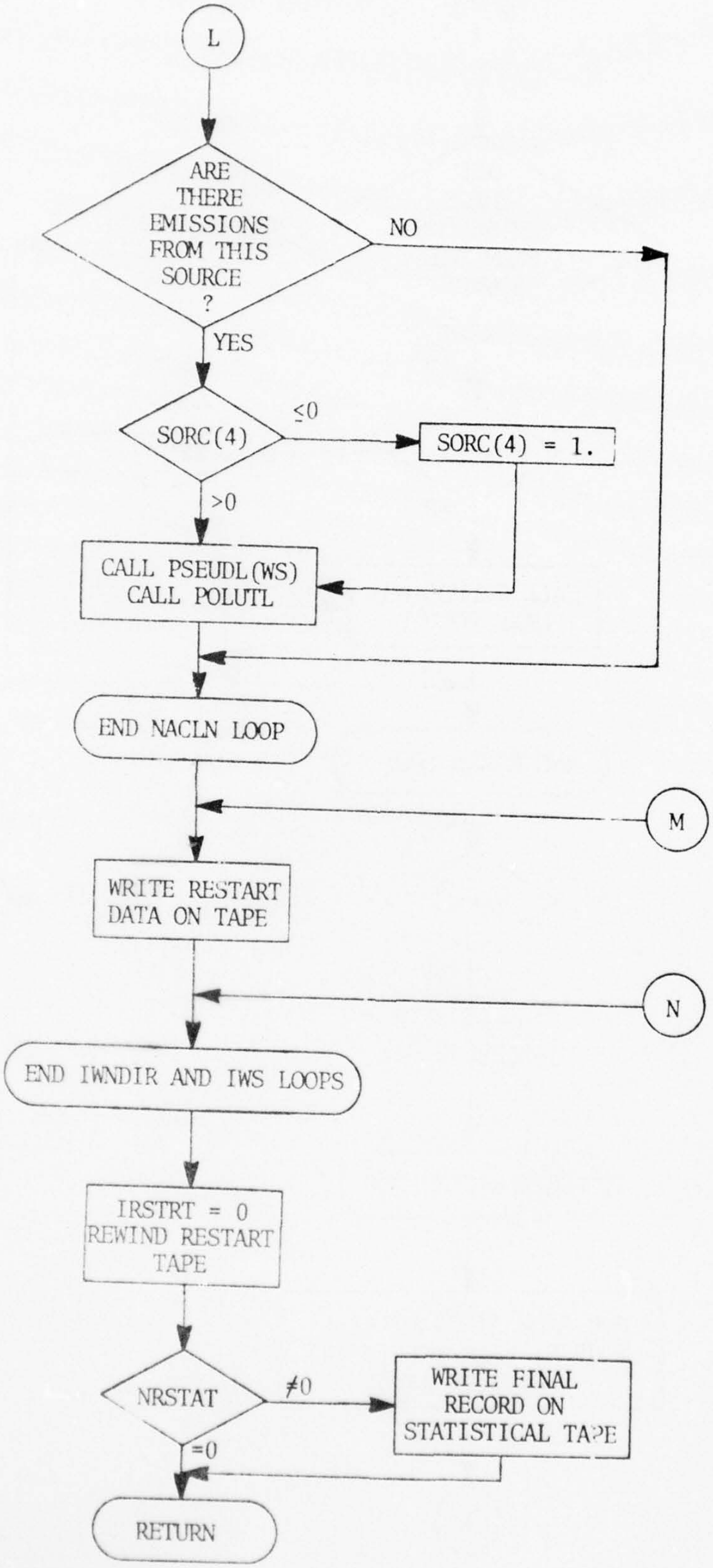

C
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NACPT LOOP
D
SET SORC VECTOR TO 0.0 AND SORC(10) = -1.
NENAR
=0
≠0
ITYPE = 1
BEGIN NENAR LOOP
FOR EACH ENVIRON AREA SOURCE, FILL SORC(1-5) AND (11-16) FROM ENAR ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NENAR LOOP
E


E
NABAR
=0
≠0
ITYPE = 2
BEGIN NABAR LOOP
FOR EACH AIRBASE AREA SOURCE, FILL SORC(1-5) AND (11-16) FROM ABAR ARRAY
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NABAR LOOP
NACAR
=0
G
≠0
ITYPE = 3
BEGIN NACAR LOOP
FOR EACH AIRCRAFT AREA SOURCE, FILL SORC(1-5) AND (11-16) FROM ACAR ARRAY
F


F
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
CALL POLUT
END NACAR LOOP
G
SORC(9) = 100.0
SORC(10) = 100.0
SORC(12) = 1.0
NENLN
=0
I
≠0
ITYPE = 1
BEGIN NENLN LOOP
FOR EACH ENVIRON LINE SOURCE, FILL SORC(1-8) AND (13-18) FROM ENLN ARRAY
H


H
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
SORC(4)
≤0
SORC(4) = 1.
>0
CALL PSEUDL(WS)
CALL POLUTL
END NENLN LOOP
I
NABLN
=0
K
≠0
ITYPE = 2
BEGIN NABLN LOOP
FOR EACH AIRBASE LINE SOURCE, FILL SORC(1-8) AND (13-18) FROM ABLN ARRAY
J


J
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
SORC(4)
≤0
SORC(4) = 1.
>0
CALL PSEUDL(WS)
CALL POLUTL
END NABLN LOOP
K
NACLN
=0
M
≠0
ITYPE = 3
BEGIN NACLN LOOP
FOR EACH AIRCRAFT LINE SOURCE, FILL SORC(1-18) FROM ACLN ARRAY
L


L
ARE THERE EMISSIONS FROM THIS SOURCE ?
NO
YES
SORC(4)
≤0
SORC(4) = 1.
>0
CALL PSEUDL(WS)
CALL POLUTL
END NACLN LOOP
M
WRITE RESTART DATA ON TAPE
N
END IWNDIR AND IWS LOOPS
IRSTRT = 0
REWIND RESTART TAPE
NRSTAT
≠0
WRITE FINAL RECORD ON STATISTICAL TAPE
=0
RETURN

```
      SUBROUTINE DIFMOD
C
C   THIS ROUTINE IS THE DRIVER FOR THE DIFFUSION MODEL.
C   FOR ALL ENVIRON, AIRBASE AND AIRCRAFT POINTS, AREA AND LINES,
C   THE SORC VECTOR IS FILLED WITH APPROPRIATE SOURCE
C   PARAMETERS AND THEN THE PROPER DIFFUSION ROUTINE IS CALLED
C
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
     . ,TEMK
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /RSTRT/ IRSTRT,IPERID,IRPRX,IRMNX,IRWSX,IRWDX
      COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
      COMMON /WDUN/ WSAVE,SUDY(6),SUDZ(6)
      DATA NINNIN/99/
      HTAERO=20.*.3048
C
C   SET RECDAT ARRAY TO 0.0
C
      DO 10 I=1,6
      DO 10 K=1,3
      DO 10 J=1,312
   10 RECDAT(K,I,J)=0.0
C
C   CHECK FOR RESTART
C
      IRWS=1
      IRWD=1
      IF(IRSTRT.EQ.0)GO TO 21
   22 READ(11,END=30)IRPR,IRMN,IRWD,IRWS,(((RECDAT(I,J,K),I=1,3),
     . J=1,NPOL),K=1,NRECEP)
      IF(IRPR.NE.IRPRX) GO TO 22
      IF(IRMN.NE.IRMNX) GO TO 22
      IF(IRWD.NE.IRWDX)GO TO 22
      IF(IRWS.NE.IRWSX)GO TO 22
      IRWS=IRWS+1
      IF(IRWS.LT.7)GO TO 21
      IRWS=1
      IRWD=IRWD+1
   21 CONTINUE
C
C   BEGIN WIND DIRECTION LOOP
C
      DO 100 IWNDIR=IRWD,16
      IRWD=1
      WD=(IWNDIR-1)*3.141597/8.0
      SINEWD=SINWD(IWNDIR)
      COSEWD=COSWD(IWNDIR)
      IWD=IWNDIR
      IFLAG=1
      DO 100 IWS=IRWS,6
      IRWS=1
      TFREQ=0.0
      DO 101 J=1,6
  101 TFREQ=TFREQ+WNDFRQ(IWS,IWNDIR,J)
```

```
      IF(TFREQ.EQ.0)GO TO 500
      WS=UU(IWS)
      IF(NCASE.EQ.0)GO TO 12
      IWDO=IWD
      DO 11 I=1,NCASE
      COMP=WS*COS(WD-WDSP(I))
      IF(COMP.LE.WSSP(I))GO TO 11
      IFLAG=1
      IWD=17+I
      IF (IWD.EQ.IWDO) IFLAG=0
      GO TO 12
   11 CONTINUE
   12 CONTINUE
C
C    DETERMINE WIND DEPENDENT SOURCES
C
      CALL ACSRCE
      IF (IFLAG.EQ.1) CALL DEPINP
      IFLAG=0
C
C    ENVIRON POINTS
C
      IF(NENPT.EQ.0)GO TO 200
      ITYPE=1
      DO 120 I=1,NENPT
      DO 121 J=1,16
  121 SORC(J)=ENPT(J,I)
      DO 321 J=6,11
      IF(SORC(J).NE.0.0)      GO TO 322
  321 CONTINUE
      GO TO 120
  322 CALL POLUT
  120 CONTINUE
C
C    AIRBASE POINTS
C
  200 IF(NABPT.EQ.0)GO TO 201
      ITYPE=2
      DO 130 I=1,NABPT
      DO 131 J=1,16
  131 SORC(J)=ABPT(J,I)
      DO 331 J=11,16
      IF(SORC(J).NE.0.0)  GO TO 332
  331 CONTINUE
      GO TO 130
  332 CALL POLUT
  130 CONTINUE
C
C    AIRCRAFT POINTS
C
  201 IF(NACPT.EQ.0)GO TO 202
      ITYPE=3
      DO 140 I=1,NACPT
      DO 141 J=1,16
  141 SORC(J)=ACPT(J,I)
      DO 341 J=11,16
      IF(SORC(J).NE.0.0 )      GO TO 342
  341 CONTINUE
      GO TO 140
  342 CALL POLUT
  140 CONTINUE
  202 CONTINUE
```

AD-A047 296 ARGONNE NATIONAL LAB ILL F/G 13/2

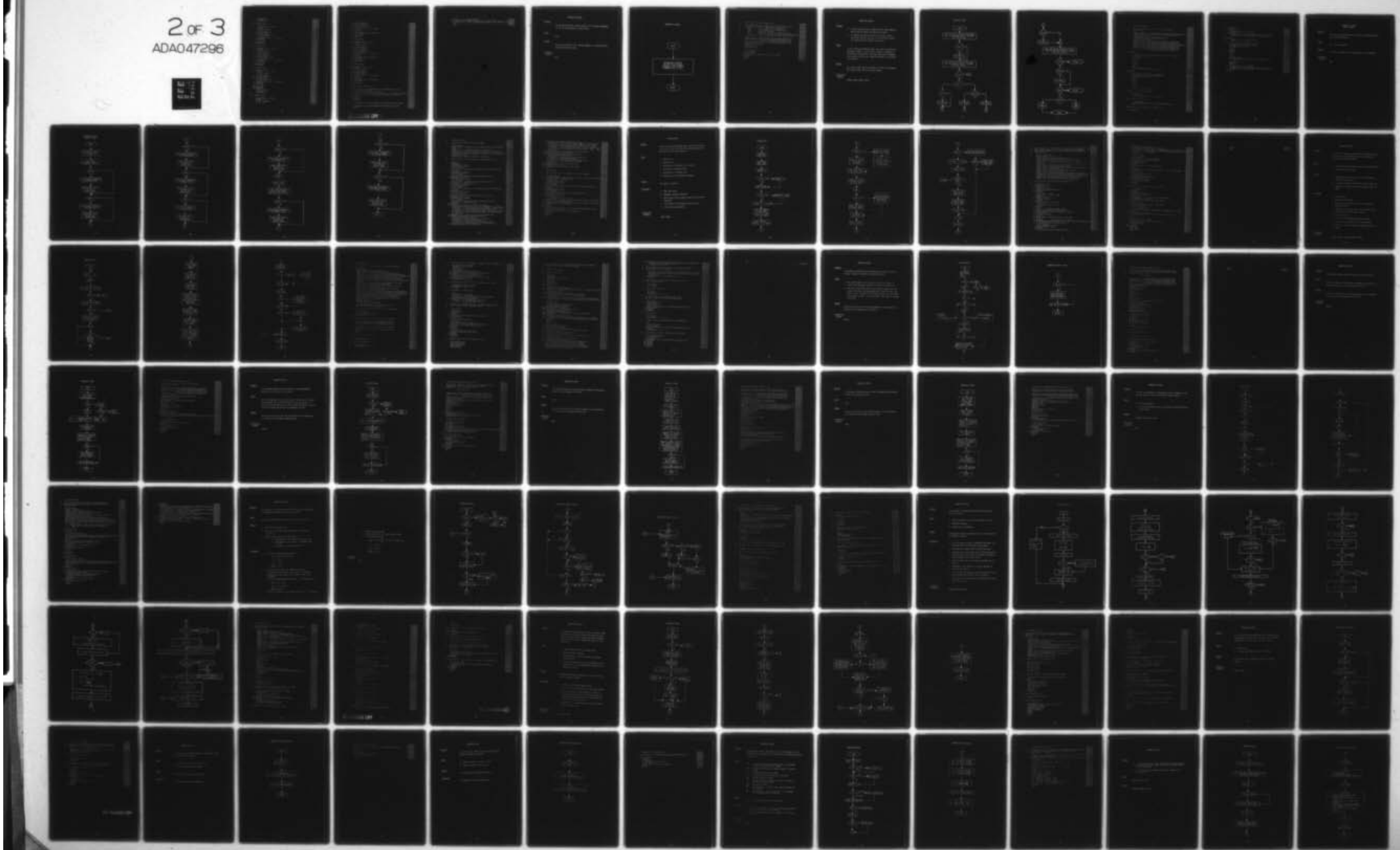
AIR QUALITY ASSESSMENT FOR AIR FORCE OPERATIONS - LONG-TERM EMI--ETC(U)

APR 77 D J BINGAMAN

UNCLASSIFIED CEEDO-TR-76-35 NL

2 OF 3

ADA047296



```
      DO 20 J=1,16
   20 SORC(J)=0.0
      SORC(10)=-1.0
C
C    ENVIRON AREAS
C
      IF(NENAR.EQ.0)GO TO 203
      ITYPE=1
      DO 122 I=1,NENAR
      DO 123 J=1,5
  123 SORC(J)=ENAR(J,I)
      DO 323 J=11,16
  323 SORC(J)=ENAR(J-5,I)
      DO 324 J=11,16
      IF(SORC(J).NE.0.0    ) GO TO 325
  324 CONTINUE
      GO TO 122
  325 CALL PCLUT
  122 CONTINUE
C
C    AIRBASE AREAS
C
  203 IF(NABAR.EQ.0)GO TO 204
      ITYPE=2
      DO 132 I=1,NABAR
      DO 133 J=1,5
  133 SORC(J)=ABAR(J,I)
      DO 333 J=11,16
  333 SORC(J)=ABAR(J-5,I)
      DO 334 J=11,16
      IF(SORC(J).NE.0.0 )    GO TO 335
  334 CONTINUE
      GO TO  132
  335 CALL PCLUT
  132 CONTINUE
C
C    AIRCRAFT AREAS
C
  204 IF(NACAR.EQ.0)GO TO 205
      ITYPE=3
      DO 142 I=1,NACAR
      DO 143 J=1,5
  143 SORC(J)=ACAR(J,I)
      DO 343 J=11,16
  343 SORC(J)=ACAR(J-5,I)
      DO 344 J=11,16
      IF(SORC(J).NE.0.0 )      GO TO 345
  344 CONTINUE
      GO TO 142
  345 CALL PCLUT
  142 CONTINUE
  205 CONTINUE
      SORC( 9)=100.0
      SORC(10)=100.0
      SORC(12)=1.
C
C    ENVIRON LINES
C
      IF(NENLN.EQ.0)GO TO 206
      ITYPE=1
      DO 125 I=1,NENLN
      DO 126 J=1,8
  126 SORC(J)=ENLN(J,I)                                                   DIFMD186
      DO 326 J=13,18                                                      DIFMD187
  326 SORC(J)=ENLN(J-4,I)                                                 DIFMD188
      DO 327 J=13,18                                                      DIFMD189
      IF(SORC(J).NE.0.0 ) GO TO 328                                       DIFMD190
  327 CONTINUE                                                            DIFMD191
      GO TO 125                                                           DIFMD192
  328 IF(SORC(4).LE.0.0) SORC(4)=1.0                                      DIFMD193
      CALL PSEUDL (WS)                                                    DIFMD194
      CALL PCLUTL                                                         DIFMD195
  125 CONTINUE                                                            DIFMD196
C                                                                         DIFMD197
C   AIRBASE LINES                                                         DIFMD198
C                                                                         DIFMD199
  206 IF(NABLN.EQ.0)GO TO 207                                             DIFMD200
      ITYPE=2                                                             DIFMD201
      DO 134 I=1,NABLN                                                    DIFMD202
      DO 135 J=1,8                                                        DIFMD203
  135 SORC(J)=ABLN(J,I)                                                   DIFMD204
      DO 336 J=13,18                                                      DIFMD205
  336 SORC(J)=ABLN(J-4,I)                                                 DIFMD206
      DO 337 J=13,18                                                      DIFMD207
      IF(SORC(J).NE.0.0) GO TO 338                                        DIFMD208
  337 CONTINUE                                                            DIFMD209
      GO TO 134                                                           DIFMD210
  338 IF(SORC(4).LE.0.0) SORC(4)=1.0                                      DIFMD211
      CALL PSEUDL (WS)                                                    DIFMD212
      CALL PCLUTL                                                         DIFMD213
  134 CONTINUE                                                            DIFMD214
C                                                                         DIFMD215
C   AIRCRAFT LINES                                                        DIFMD216
C                                                                         DIFMD217
  207 IF(NACLN.EQ.0)GO TO 208                                             DIFMD218
      ITYPE=3                                                             DIFMD219
      DO 144 I=1,NACLN                                                    DIFMD220
      DO 145 J=1,18                                                       DIFMD221
  145 SORC(J)=ACLN(J,I)                                                   DIFMD222
      DO 346 J=13,18                                                      DIFMD223
      IF(SORC(J).NE.0.0  ) GO TO 347                                      DIFMD224
  346 CONTINUE                                                            DIFMD225
      GO TO 144                                                           DIFMD226
  347 IF(SORC(4).LE.0.0) SORC(4)=1.0                                      DIFMD227
      CALL PSEUDL (WS)                                                    DIFMD228
      CALL PCLUTL                                                         DIFMD229
  144 CONTINUE                                                            DIFMD230
C                                                                         DIFMD231
C   WRITE RESTART DATA ON 11                                              DIFMD232
C                                                                         DIFMD233
  208 WRITE(11)IPERID,IMONTH,IWNDIR,IWS,(((RECDAT(I,J,K),I=1,3),          DIFMD234
     . J=1,NPOL),K=1,NRECEP)                                              DIFMD235
      WRITE(6,302)IPERID,IMONTH,IWS,IWNDIR                                DIFMD236
  302 FORMAT(70H RESTART DATA HAS BEEN WRITTEN -- TO RESTART FROM THIS PDIFMD237
     .OINT SET IRPP=  ,I6,6H IRMN=,I6,6H IRWS=,I6,6H IRWD=,I6)            DIFMD238
  500 CONTINUE                                                            DIFMD239
  100 CONTINUE                                                            DIFMD240
      IRSTRT=0                                                            DIFMD241
      REWIND 11                                                           DIFMD242
C                                                                         DIFMD243
C   IF STATISTICAL OPTION IS CHOSEN WRITE A FINAL RECORD ON NSTAPE        DIFMD244
C                                                                         DIFMD245
      IF (NRSTAT.NE.0) WRITE (NSTAPE) NINNIN,(SORC(J),J=1,14)             DIFMD246
      RETURN                                                              DIFMD247
```

```
 30 WRITE(6,303) IRPR,IRMN,IRWD,IRWS                                   DIFMD248
303 FORMAT(30H1LAST RESTART RECORD IS IRPR =,I4,7H IRMN =,I4,          DIFMD249
   . 7H IRWD =,I4,7H IRWS =,I4/35H0CHANGE INPUT CARD AND RESUBMIT JOB)DIFMD250
    STOP                                                               DIFMD251
    END                                                                DIFMD252
```

SUBROUTINE EMISAR

Purpose:

To add emissions from a given activity to all others contained in the specified geometric area or line.

Input:

None

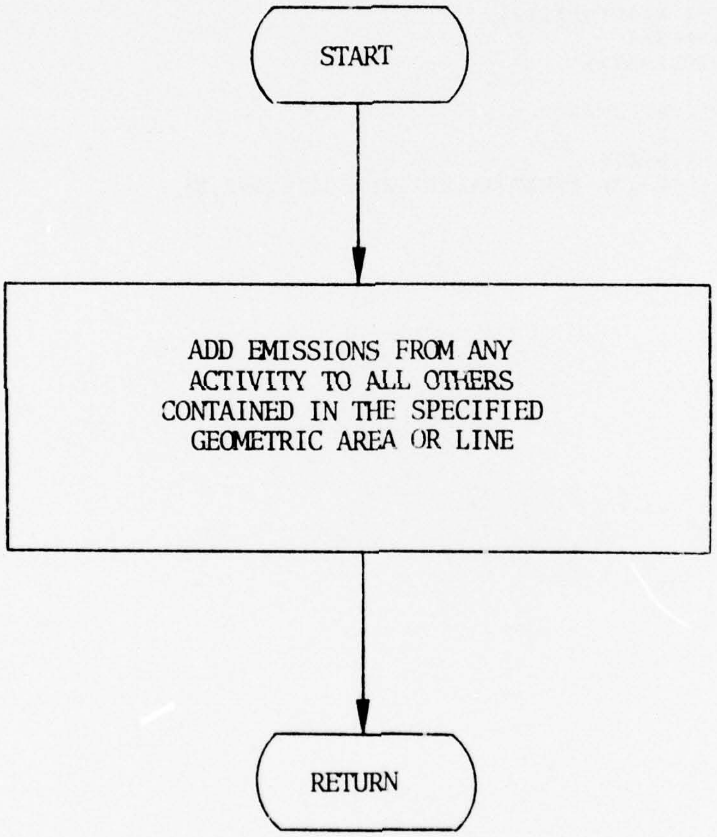
Output:

The array specified in the calling sequence is filled with the accumulated emission data.

Subroutines Called:

None

SUBROUTINE EMISAR


START
ADD EMISSIONS FROM ANY ACTIVITY TO ALL OTHERS CONTAINED IN THE SPECIFIED GEOMETRIC AREA OR LINE
RETURN

```
      SUBROUTINE EMISAR(MAXN,ARRAY,I1,I2)
C
C     THIS ROUTINE ACCUMULATES EMISSIONS FROM ANY ACTIVITY WITH
C     OTHERS CONTAINED IN THE SAME AIRBASE AREA OR LINE.
C        MAXN        = NO. OF SOURCES IN AN ACTIVITY
C        ARRAY       = SPECIFIED AREA OR LINE OUTPUT ARRAY
C        I1,I2       = DIMENSIONS OF ARRAY
C        NSRCE       = POINTER TO LOCATION OF SOURCES IN SORCE
C        LOC1        = POINTER TO LOCATION OF LIST OF EMISSIONS IN ARRAY
C        SORCE(2,N)  = POINTER TO LOCATION OF SOURCE AREA OR LINE
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                    ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
C
      DO 10 N=LSRCE,NSRCE
      J=SORCE(2,N)
      DO 10 I=1,NPLTS
      ARRAY(I+LOC1,J)=ARRAY(I+LOC1,J)+SORCE(I+2,N)
   10 CONTINUE
      RETURN
      END
```

SUBROUTINE ENARAY

Purpose:

1. To read from the master source tape all data needed to define environ point, area and line sources.
2. To compute the emission rates due to point sources, stationary, mobile, land use or combined area sources and roadway and non-roadway line sources.

Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those land use or combined area source activities not using the default of a uniform distribution.

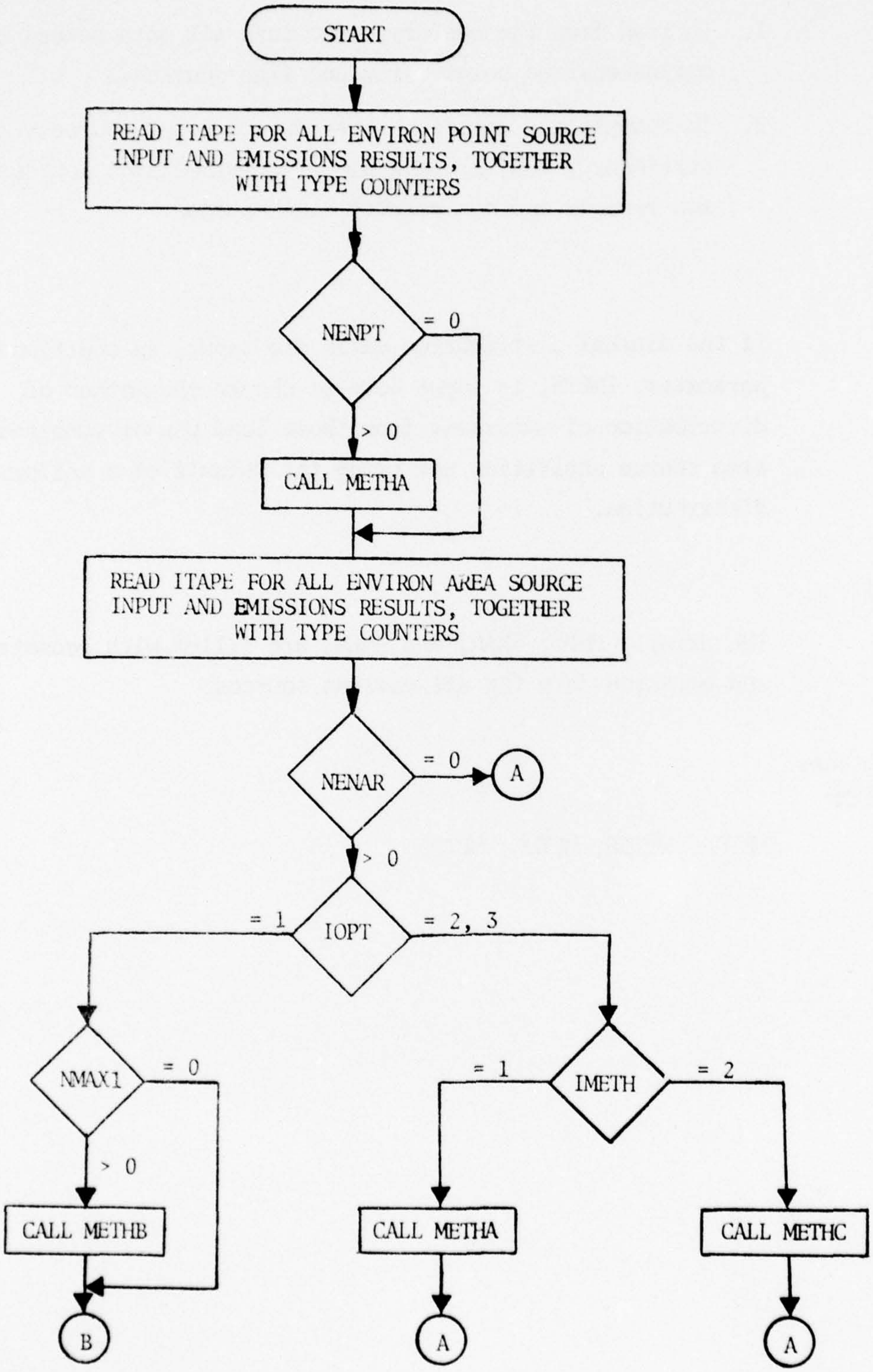
Output:

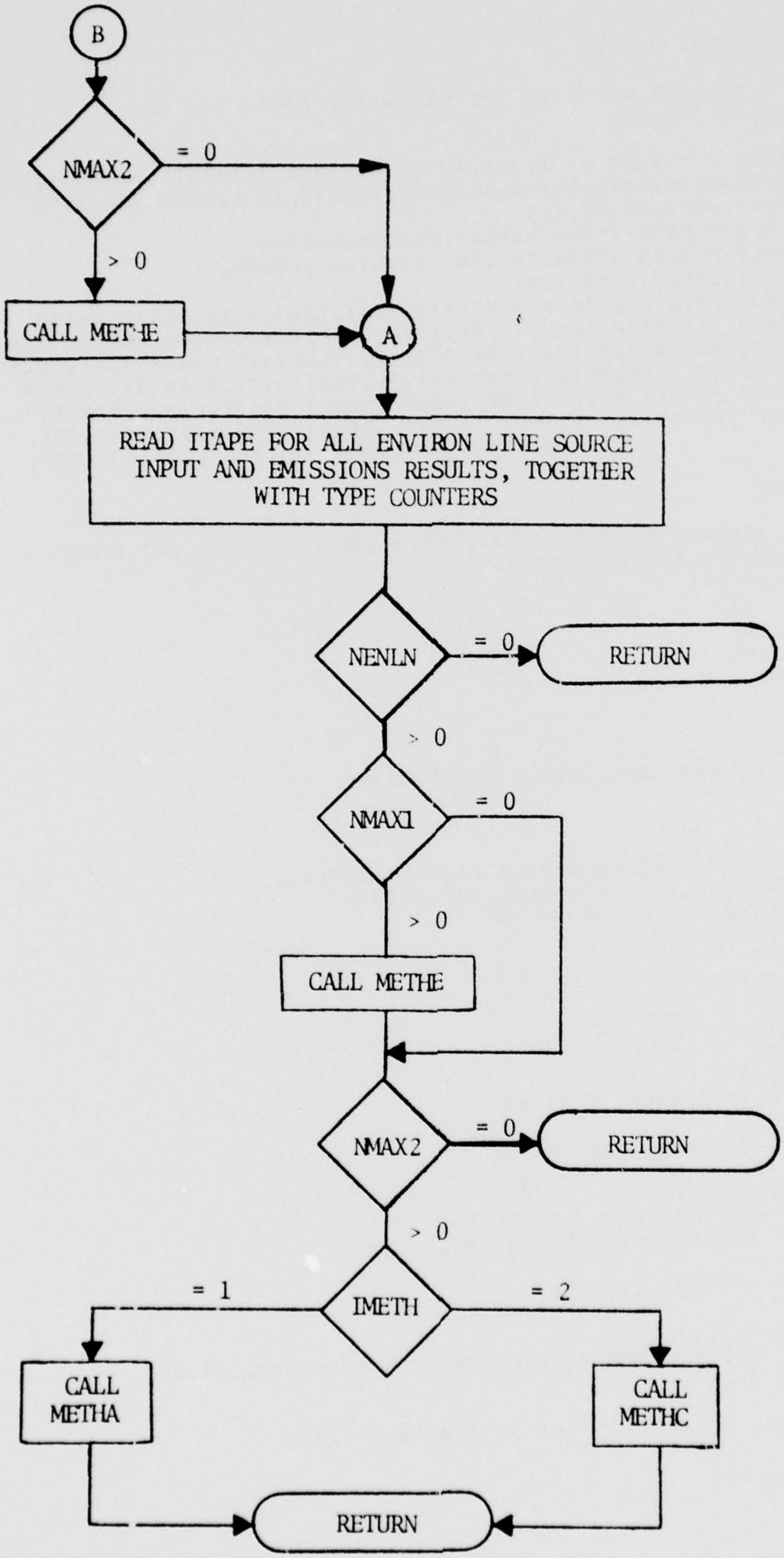
The arrays, ENPT, ENAR, and ENLN, are filled with geometry and emission data for all environ sources.

Subroutines Called:

METHA, METHB, METHC, METHE

SUBROUTINE ENARAY


START
READ ITAPE FOR ALL ENVIRON POINT SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS
NENPT
= 0
> 0
CALL METHA
READ ITAPE FOR ALL ENVIRON AREA SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS
NENAR
= 0
A
> 0
IOPT
= 1
= 2, 3
NMAX1
= 0
> 0
CALL METHB
B
IMETH
= 1
= 2
CALL METHA
A
CALL METHC
A


B
NMAX2
= 0
> 0
CALL METHE
A
READ ITAPE FOR ALL ENVIRON LINE SOURCE INPUT AND EMISSIONS RESULTS, TOGETHER WITH TYPE COUNTERS
NENLN
= 0
RETURN
> 0
NMAX1
= 0
> 0
CALL METHE
NMAX2
= 0
RETURN
> 0
IMETH
= 1
= 2
CALL METHA
CALL METHC
RETURN

```
      SUBROUTINE ENARAY
C
C     THIS ROUTINE COMPUTES THE EMISSSION RATES FOR ALL
C     ENVIRON SOURCES
C
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/MONMET/TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,
     .  JSTAB,HLID,TEMF,TEMK
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                       ABPT(16,150),ABAR(11,100),ABLN(14,100)
C
C****POINTS
C
      READ(ITAPE) NENPT,NTOT,((SORCE(I,N),I=1,NTOT),N=1,NENPT)
      IF (NENPT.EQ.0) GO TO 100
      ICLASS=201
      LOC1=10
      LOC2=11
      NGEOM=9
      I1=16
      I2=100
      IPT=1
      NSRCE=0
      CALL METHA(NENPT,ENPT,I1,I2,ICLASS)
C
C****AREAS
C
  100 READ(ITAPE) NENAR,NTOT,IOPT,NMAX1,NMAX2,
     1  ((SORCE(I,N),I=1,NTOT),N=1,NENAR)
      IF (NENAR.EQ.0) GO TO 300
      LOC1=5
      LOC2=7
      NGEOM=5
      IPT=0
      I1=11
      I2=100
      NSRCE=0
      GO TO (110,120,130),IOPT
C
C****OPTION 1  STATIONARY AREAS
C
  110 ICLASS=202
      IF (NMAX1.GT.0)
     1 CALL METHB(NMAX1,ENAR,I1,I2,ICLASS)
C
C****OPTION 1  MOBILE AREAS
C
      IF (NMAX2.GT.0)
     1 CALL METHE(NMAX2,ENAR,CVENMO,CVENDY,CVENHR,I1,I2)
      GO TO 300
C
C****OPTION 2 OR 3  LAND USE OR COMBINED AREAS
C
  120 ICLASS=203
```

```
      GO TO 200                                                         ENARY062
  130 ICLASS=204                                                        ENARY063
  200 IMETH=1                                                           ENARY064
      IF (JFLAG.EQ.0) READ 201,IMETH                                    ENARY065
  201 FORMAT(I4)                                                        ENARY066
      IF (IMETH.EQ.1) CALL METHA(NMAX1,ENAR,I1,I2,ICLASS)               ENARY067
      IF (IMETH.EQ.2) CALL METHC(NMAX1,ENAR,I1,I2,ICLASS)               ENARY068
C                                                                       ENARY069
C****LINES                                                              ENARY070
C       NMAX1 = NO. OF  ROADWAY LINES                                   ENARY071
C       NMAX2 = NO. OF NON-ROADWAY LINES                                ENARY072
C                                                                       ENARY073
  300 READ(ITAPE) NENLN,NTOT,NMAX1,NMAX2,                               ENARY074
     1  ((SORCE(I,N),I=1,NTOT),N=1,NENLN)                               ENARY075
      IF (NENLN.EQ.0) GO TO 400                                         ENARY076
      LOC1=8                                                            ENARY077
      LOC2=10                                                           ENARY078
      NGEOM=8                                                           ENARY079
      NSRCE=0                                                           ENARY080
      I1=14                                                             ENARY081
      I2=20                                                             ENARY082
      IFT=0                                                             ENARY083
      IF (NMAX1.GT.0)                                                   ENARY084
     1 CALL METHE(NMAX1,ENLN,CVENMO,CVENDY,CVENHR,I1,I2)                ENARY085
C                                                                       ENARY086
      IF (NMAX2.EQ.0) GO TO 400                                         ENARY087
      ICLASS=206                                                        ENARY088
      IMETH=1                                                           ENARY089
      IF (JFLAG.EQ.0) READ 201,IMETH                                    ENARY090
      IF (IMETH.EQ.1) CALL METHA(NMAX2,ENLN,I1,I2,ICLASS)               ENARY091
      IF (IMETH.EQ.2) CALL METHC(NMAX2,ENLN,I1,I2,ICLASS)               ENARY092
  400 RETURN                                                            ENARY093
      END                                                               ENARY094
```

SUBROUTINE INDINP
ENTRY DEPINP

Purpose:

To print the input parameters for both wind independent and wind dependent sources.

Input:

All source parameters.

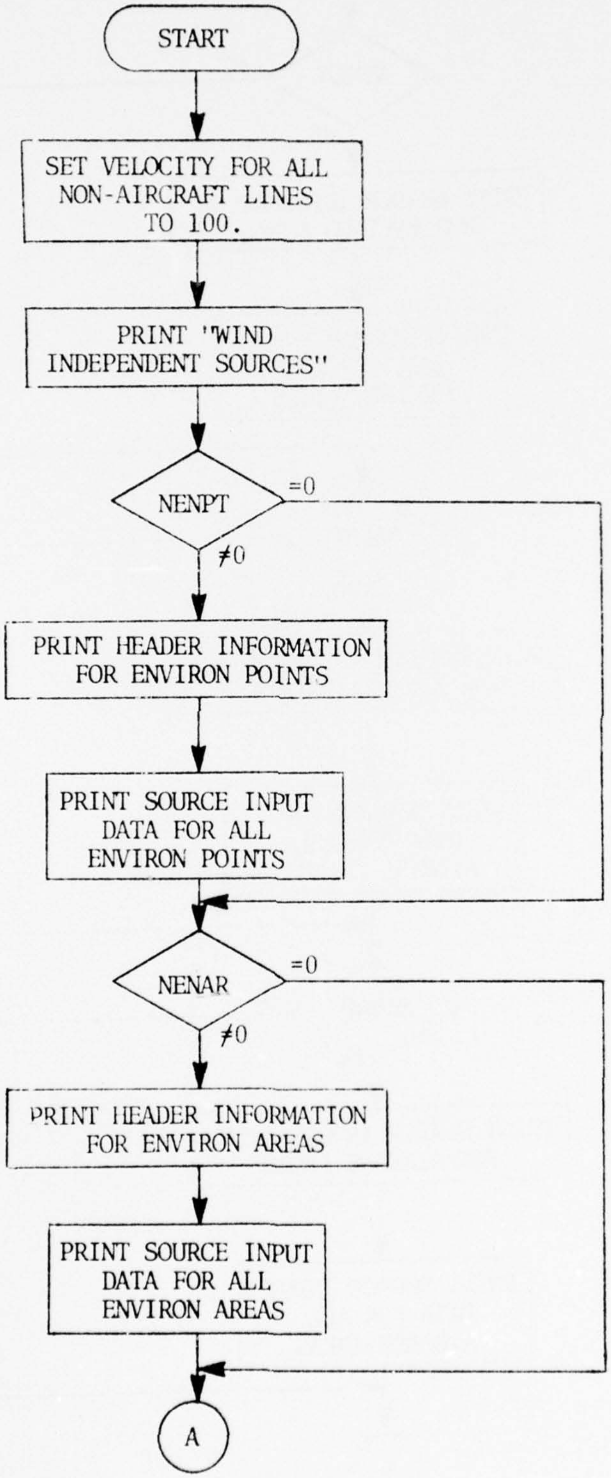
Output:

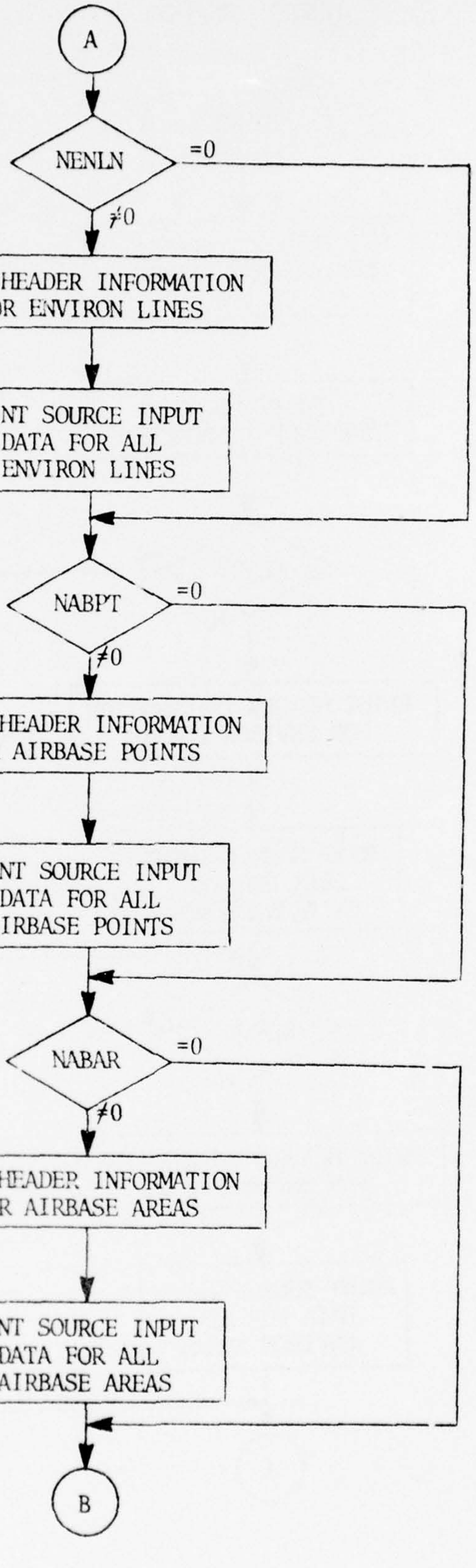
All source parameters with appropriate title information.

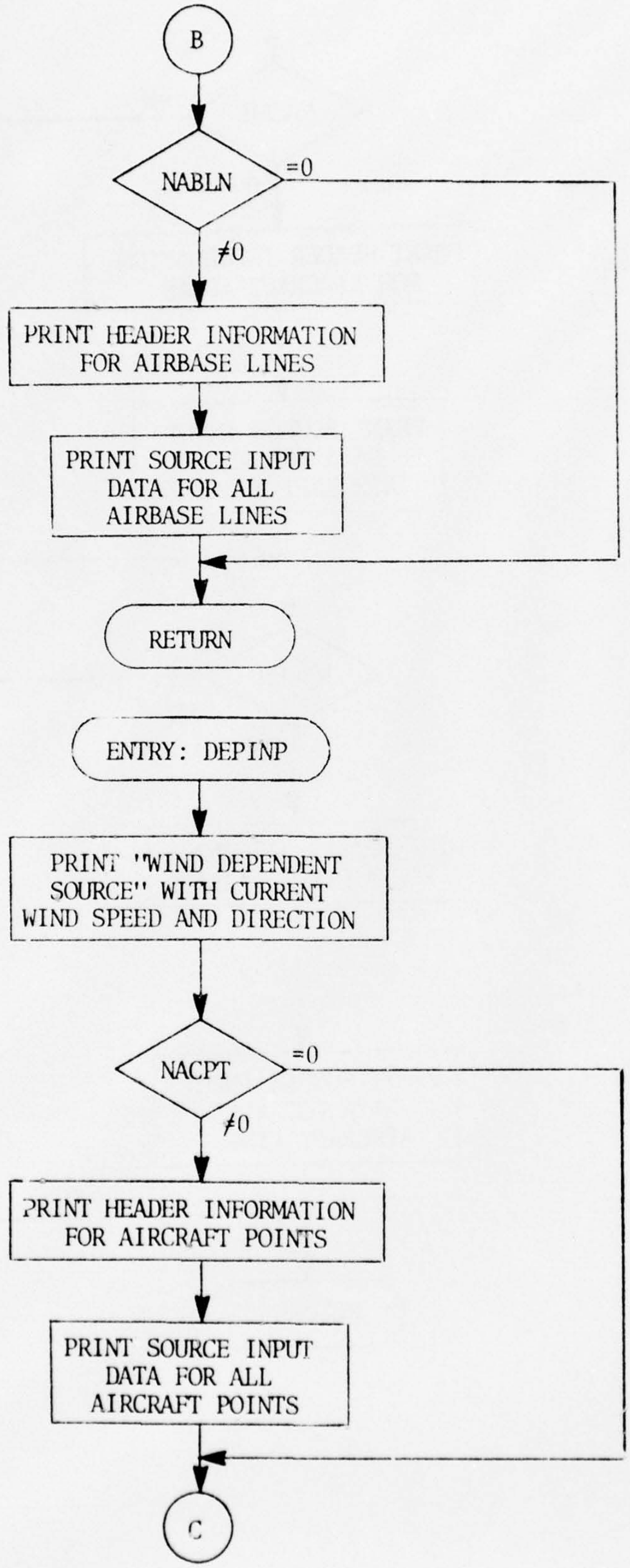
Subroutines Called:

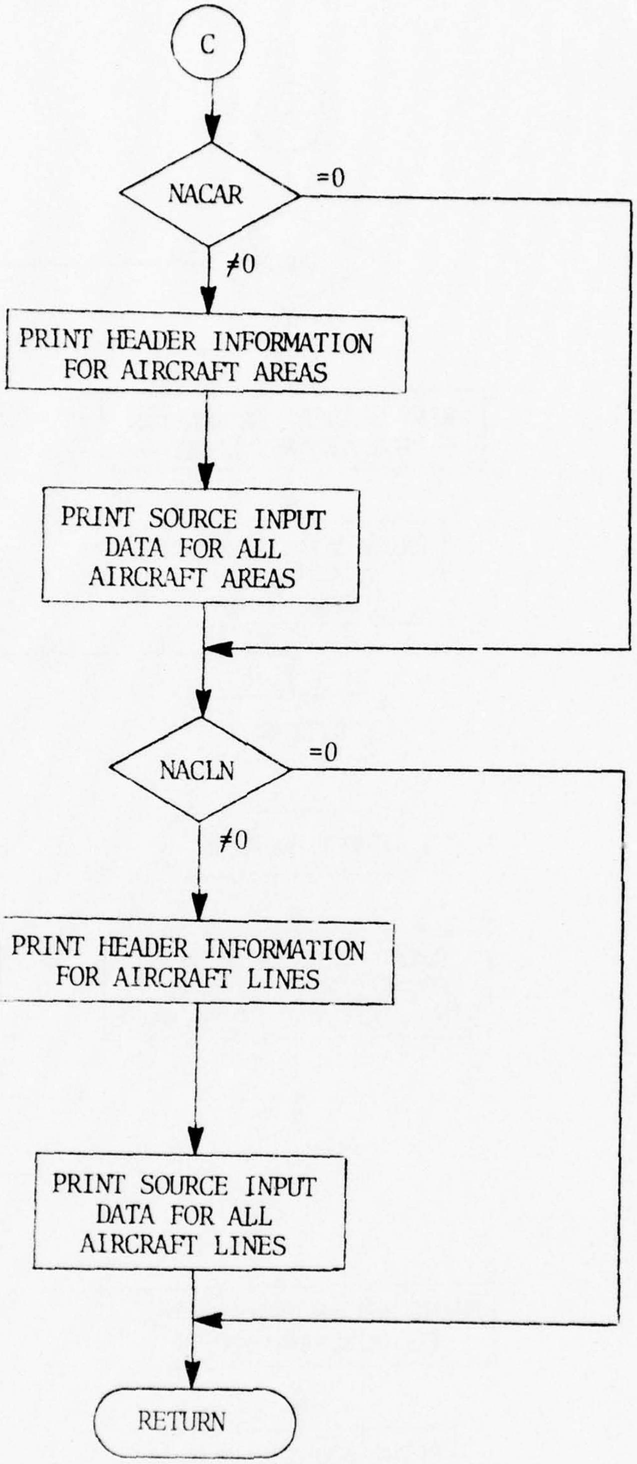
None

SUBROUTINE INDINP
(ENTRY: DEPINP)


START
SET VELOCITY FOR ALL NON-AIRCRAFT LINES TO 100.
PRINT "WIND INDEPENDENT SOURCES"
NENPT
=0
≠0
PRINT HEADER INFORMATION FOR ENVIRON POINTS
PRINT SOURCE INPUT DATA FOR ALL ENVIRON POINTS
NENAR
=0
≠0
PRINT HEADER INFORMATION FOR ENVIRON AREAS
PRINT SOURCE INPUT DATA FOR ALL ENVIRON AREAS
A


A
NENLN
=0
≠0
PRINT HEADER INFORMATION FOR ENVIRON LINES
PRINT SOURCE INPUT DATA FOR ALL ENVIRON LINES
NABPT
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE POINTS
PRINT SOURCE INPUT DATA FOR ALL AIRBASE POINTS
NABAR
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE AREAS
PRINT SOURCE INPUT DATA FOR ALL AIRBASE AREAS
B


B
NABLN
=0
≠0
PRINT HEADER INFORMATION FOR AIRBASE LINES
PRINT SOURCE INPUT DATA FOR ALL AIRBASE LINES
RETURN
ENTRY: DEPINP
PRINT "WIND DEPENDENT SOURCE" WITH CURRENT WIND SPEED AND DIRECTION
NACPT
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT POINTS
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT POINTS
C


C
NACAR
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT AREAS
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT AREAS
NACLN
=0
≠0
PRINT HEADER INFORMATION FOR AIRCRAFT LINES
PRINT SOURCE INPUT DATA FOR ALL AIRCRAFT LINES
RETURN

```
      SUBROUTINE INDINP
C
C     THIS ROUTINE PRINTS ALL THE SOURCE INPUT
C
      REAL*8 SCRNAM
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
     . ,TEMK
      COMMON /PERIOD/ IMO,NODAYS,IDY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      DIMENSION SCRNAM(3)
      DATA SCRNAM / 8HENVIRON ,8HAIRBASE ,8HAIRCRAFT /
C
C     AT THIS ENTRY ALL WIND INDEPENDENT SOURCES ARE PRINTED
C
      ENLNA=1.0
      WRITE(6,200)
      IF(NENPT.EQ.0)GO TO 11
      WRITE(6,100) SCRNAM(1)
      DO 1 I=1,NENPT
    1 WRITE(6,101)I,(ENPT(J,I),J=1,4),(ENPT(J,I),J=6,8),(ENPT(J,I),
     . J=10,15)
   11 IF (NENAR.EQ.0)GO TO 12
      WRITE(6,110) SORNAM(1)
      DO 2 I=1,NENAR
    2 WRITE(6,111)I,(ENAR(J,I),J=1,4),(ENAR(J,I),J=6,10)
   12 IF(NENLN.EQ.0)GO TO 13
      WRITE(6,1200)  SORNAM(1)
      DO 5 I=1,NENLN
      WRITE(6,1211)  I,(ENLN(J,I),J=1,4),(ENLN(J,I),J=9,13)
    5 WRITE(6,1222)  (ENLN(J,I),J=6,8)
   13 IF (NABPT.EQ.0)GO TO 14
      WRITE(6,100)SCRNAM(2)
      DO 3 I=1,NABPT
    3 WRITE(6,101)I,(ABPT(J,I),J=1,4),(ABPT(J,I),J=6,8),(ABPT(J,I),
     . J=10,15)
   14 IF(NABAR.EQ.0)GO TO 15
      WRITE(6,110)SCRNAM(2)
      DO 4 I=1,NABAR
    4 WRITE(6,111)I,(ABAR(J,I),J=1,4),(ABAR(J,I),J=6,10)
   15 IF(NABLN.EQ.0)GO TO 16
      WRITE(6,1200)  SCRNAM(2)
      DO 6 I=1,NABLN
      WRITE(6,1211)  I,(ABLN(J,I),J=1,4),(ABLN(J,I),J=9,13)
    6 WRITE(6,1222)  (ABLN(J,I),J=6,8)
   16 CONTINUE
  100 FORMAT(1H0,A8,14H POINT SOURCES/1X,119(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,11X,22HI   STACK PARAMETERS   I,4X,1HI/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,12HWIDTH I TEMP,4X,3HVEL,
     . 3X,11HDIAM I PR I,13X,28HEMISSIONS(MICROGRAMS/SECOND)/
     . 1X,14HNUMBER I   (KM),5X,4H(KM),4X,25H(M)     (M)   I(DEG K)  (M/S),
     . 3X,10H(M)  IFLAGI,4X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/
     . 1X,119(1H-))
  101 FORMAT(I6,1X,2F9.2,2F7.1,F7.0,2F7.1,F4.0,5(1PE11.3))
  110 FORMAT(1H0,A8,13H AREA SOURCES/1X,94(1H-)/
     . 8X,1HI,11X,8HGEOMETRY,11X,1HI/
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,6HSIDE I,
     . 14X,28HEMISSIONS(MICROGRAMS/SECOND)/
     . 1X,14HNUMBER I   (KM),5X,4H(KM),4X,3H(M),4X,5H(M) I,
     . 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/1X,94(1H-))
  111 FORMAT(I6,1X,2F9.2,2F7.1,5(1PE11.3))
  120 FORMAT(1H0,A8,13H LINE SOURCES/1X,123(1H-)/                        INDIP062
     . 8X,1HI,11X,8HGEOMETRY,11X,1HI,10X,1HI,54X,15HI AIRCRAFT ONLY/     INDIP063
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,18HWIDTH I VELOCITY I,      INDIP064
     . 13X,28HEMISSIONS(MICROGRAMS/SECOND),13X,15HI LENGTH   TIME/       INDIP065
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,23H(M)    (M)  I (KM/HR)  I,    INDIP066
     . 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2,3X,15HI  (KM)     (HR)/ INDIP067
     . 1X,123(1H-))                                                      INDIP068
  121 FORMAT(I6,1X,2F9.2,2F7.1,6(1PE11.3),0PF7.2,1PE11.3)                INDIP069
  122 FORMAT(7X,2F9.2,F7.1,7X,1PE11.3)                                   INDIP070
 1200 FORMAT(1H0,A8,13H LINE SOURCES/1X,96(1H-)/                         INDIP071
     . 8X,1HI,11X,8HGEOMETRY,12X,1HI/                                    INDIP072
     . 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,7HWIDTH I,                  INDIP073
     . 23X,28HEMISSIONS(MICROGRAMS/SECOND)/                              INDIP074
     . 1X,14HNUMBER I  (KM),5X,4H(KM),4X,3H(M),4X,6H(M)   I,             INDIP075
     . 6X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/                        INDIP076
     . 1X,96(1H-))                                                       INDIP077
 1211 FORMAT(I6,1X,2F9.2,2F7.1,2X,5(1PE11.3))                            INDIP078
 1222 FORMAT(7X,2F9.2,F7.1)                                              INDIP079
  200 FORMAT(25H0WIND INDEPENDENT SOURCES/1H0)                           INDIP080
      RETURN                                                             INDIP081
C                                                                        INDIP082
      ENTRY DEPINP                                                       INDIP083
C                                                                        INDIP084
C     AT THIS ENTRY ALL WIND DEPENDENT SOURCES ARE PRINTED               INDIP085
C                                                                        INDIP086
      WRITE(6,300)WS,WD                                                  INDIP087
  300 FORMAT(1H1,'WIND DEPENDENT SOURCES FOR',F8.4,' MPS WIND SPEED AND'INDIP088
     . ,F8.4,' RADIANS WIND DIRECTION')                                  INDIP089
      IF (IFLAG.EQ.0) GO TO 18                                           INDIP090
      IF (NACPT.EQ.0) GO TO 17                                           INDIP091
      WRITE(6,100) SORNAM(3)                                             INDIP092
      DO 7 I=1,NACPT                                                     INDIP093
    7 WRITE(6,101)I,(ACPT(J,I),J=1,4),(ACPT(J,I),J=6,8),(ACPT(J,I),      INDIP094
     . J=11,15)                                                          INDIP095
   17 IF(NACAR.EQ.0) GO TO 18                                            INDIP096
      WRITE(6,110) SORNAM(3)                                             INDIP097
      DO 8 I=1,NACAR                                                     INDIP098
    8 WRITE(6,111)I,(ACAR(J,I),J=1,4),(ACAR(J,I),J=6,10)                 INDIP099
   18 IF(NACLN.EQ.0) GO TO 19                                            INDIP100
      WRITE(6,120) SORNAM(3)                                             INDIP101
      DO 9 I=1,NACLN                                                     INDIP102
      IF (ACLN(9,I).NE.1.0) GO TO 1987                                   INDIP103
      WRITE(6,1219) I,(ACLN(J,I),J=1,4),(ACLN(J,I),J=13,17),ACLN(11,I)   INDIP104
      WRITE(6,1229) (ACLN(J,I),J=6,8)                                    INDIP105
 1219 FORMAT(I6,1X,2F9.2,2F7.1,4X,3HN/A,4X,5(1PE11.3),0PF7.1,4X,3HN/A)   INDIP106
 1229 FORMAT(7X,2F9.2,F7.1,11X,3HN/A)                                    INDIP107
      GO TO 9                                                            INDIP103
 1987 CONTINUE                                                           INDIP109
      WRITE(6,121)I,(ACLN(J,I),J=1,4),ACLN(9,I),(ACLN(J,I),J=13,17),     INDIP110
     . ACLN(11,I),ACLN(12,I)                                             INDIP111
      WRITE(6,122)  (ACLN(J,I),J=6,8),ACLN(10,I)                         INDIP112
    9 CONTINUE                                                           INDIP113
   19 CONTINUE                                                           INDIP114
      RETURN                                                             INDIP115
      END                                                                INDIP116
```

PROGRAM MAIN

Purpose:

To read the general problem input, set up the receptor grid, call a routine to read the master emission file and then call the long-term model.

Input:

1. Problem title
2. Restart data
3. Definition of pollutants to be output
4. Description of special cases
5. Description of receptor grid
6. Description of statistical receptors

Output:

All input is printed.

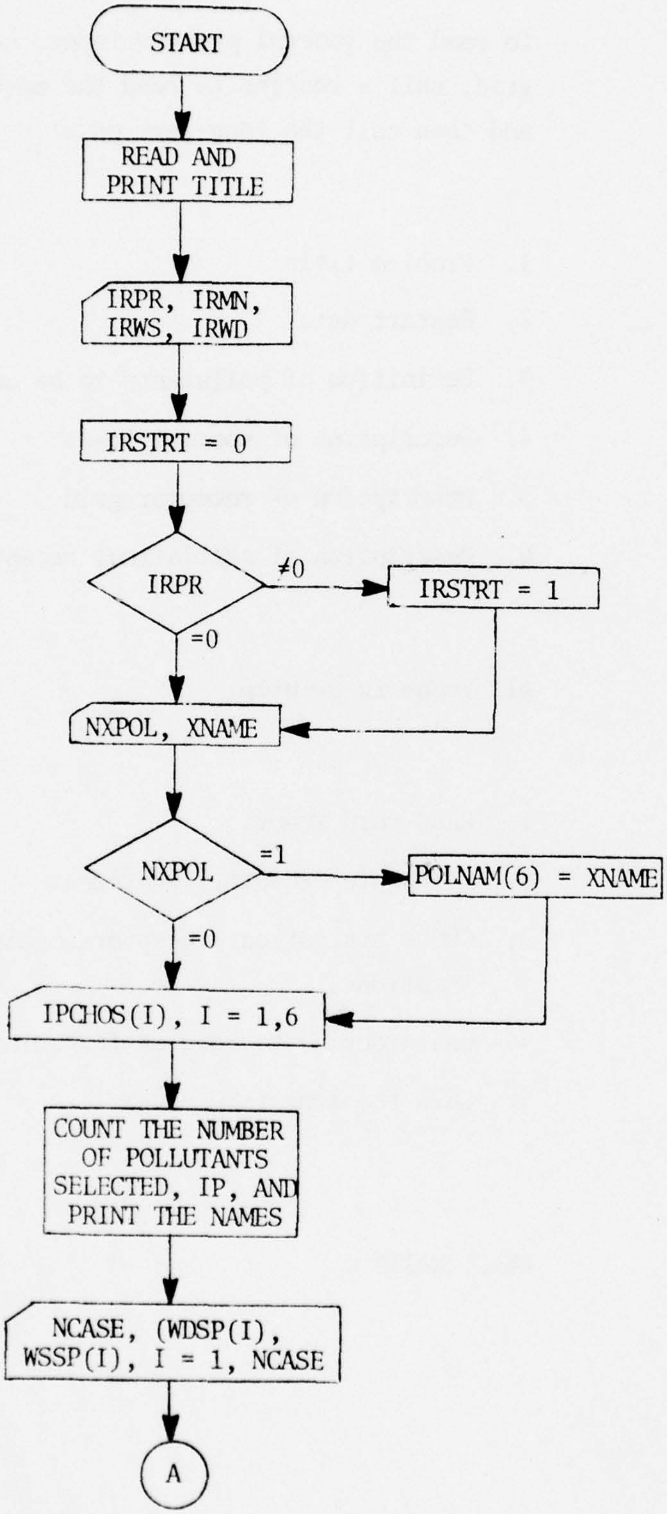
Procedure:

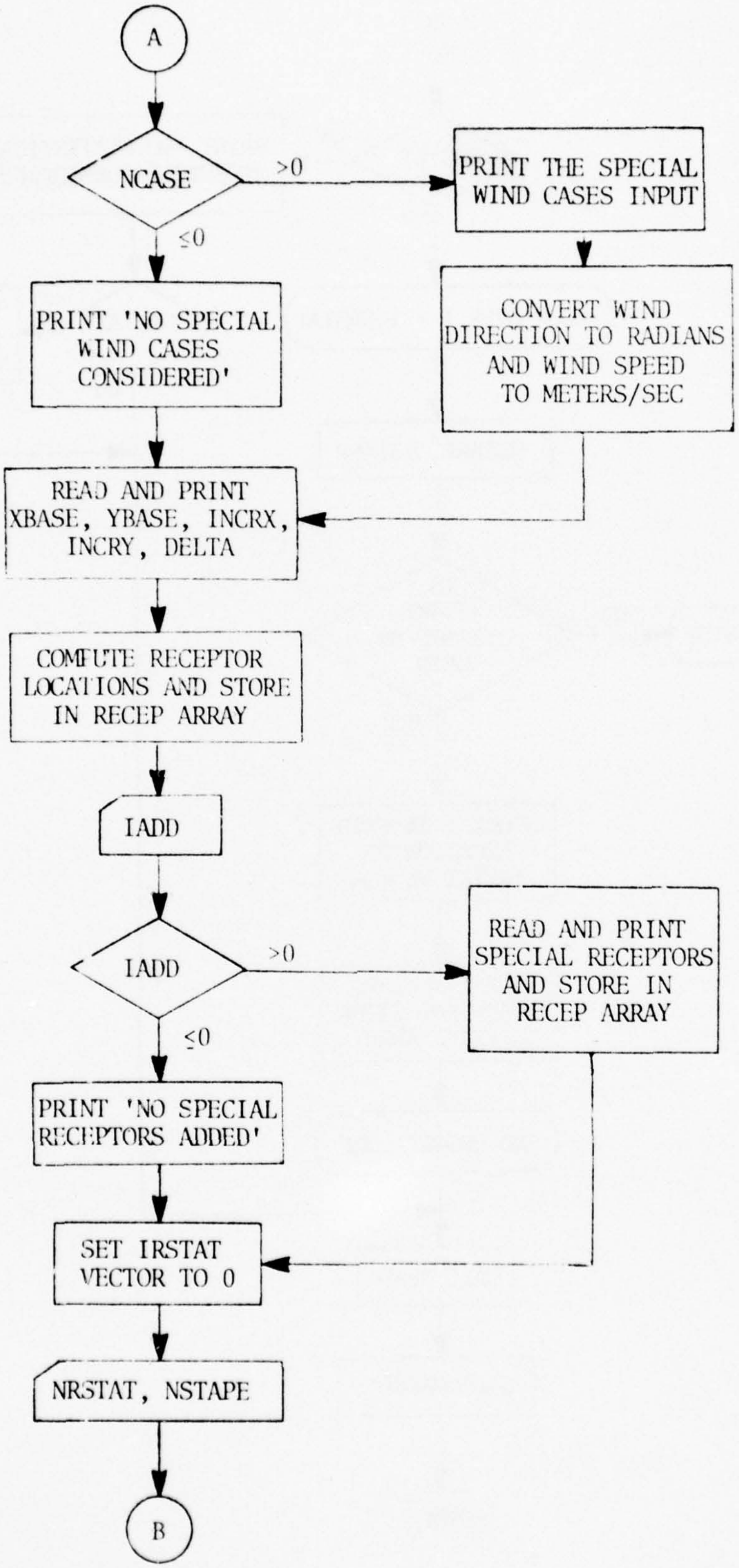
1. Read card input.
2. Calculate receptor locations.
3. Check statistical receptors against the receptor locations.
4. Call routine to read master emission file.
5. Call the long-term model.

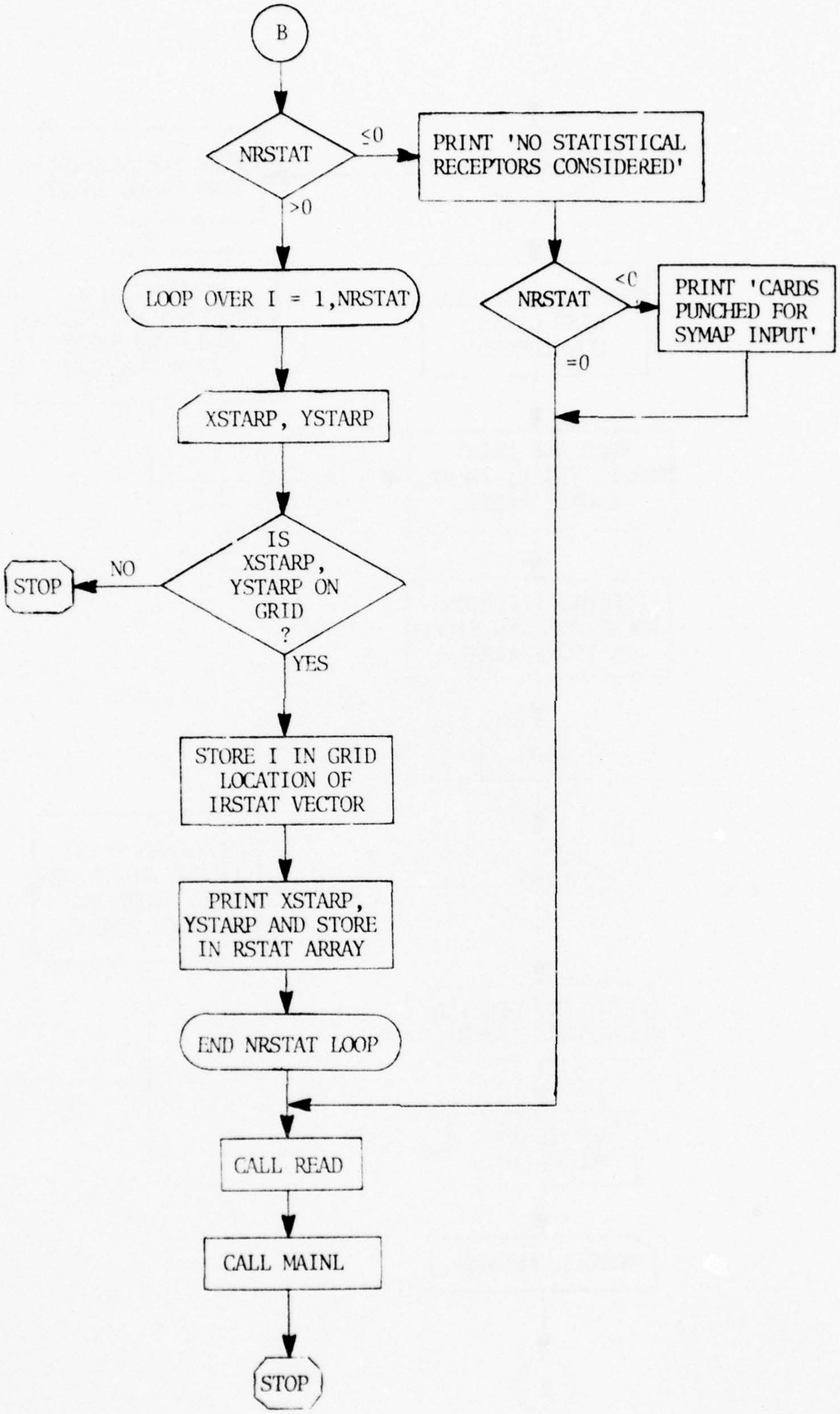
Subroutines Called:

READ, MAINL

PROGRAM MAIN




A
NCASE
>0
≤0
PRINT THE SPECIAL WIND CASES INPUT
PRINT 'NO SPECIAL WIND CASES CONSIDERED'
CONVERT WIND DIRECTION TO RADIANS AND WIND SPEED TO METERS/SEC
READ AND PRINT XBASE, YBASE, INCRX, INCRY, DELTA
COMPUTE RECEPTOR LOCATIONS AND STORE IN RECEP ARRAY
IADD
IADD
>0
≤0
READ AND PRINT SPECIAL RECEPTORS AND STORE IN RECEP ARRAY
PRINT 'NO SPECIAL RECEPTORS ADDED'
SET IRSTAT VECTOR TO 0
NRSTAT, NSTAPE
B


B
NRSTAT
≤0
PRINT 'NO STATISTICAL RECEPTORS CONSIDERED'
>0
LOOP OVER I = 1,NRSTAT
NRSTAT
<0
PRINT 'CARDS PUNCHED FOR SYMAP INPUT'
=0
XSTARP, YSTARP
IS XSTARP, YSTARP ON GRID ?
NO
STOP
YES
STORE I IN GRID LOCATION OF IRSTAT VECTOR
PRINT XSTARP, YSTARP AND STORE IN RSTAT ARRAY
END NRSTAT LOOP
CALL READ
CALL MAINL
STOP

```
C
C
C    THIS PROGRAM IS THE MAIN DRIVER ROUTINE WHICH READS IN RECEPTOR AND
C    OTHER GENERAL DATA, CALLS SUBROUTINE READ TO READ THE MASTER
C    SOURCE EMISSION TAPE, AND THE DIRECTS CONTROL TO MAINL FOR THE
C    LONG TERM MODEL
C
      REAL*8 POLNAM,XNAME
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
     . ,TEMK,UA
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)
      COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /RSTRT/ IRSTRT,IPERID,IRPR,IRMN,IRWS,IRWD
      COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
      COMMON /SRCE/ NPOL,NEMPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /TITL/ POLNAM(6),TITLE1(20),IPCHOS(6),NXPOL,IP
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
C
C    READ AND PRINT RECEPTOR AND OTHER GENERAL INPUT
C
    1 READ(5,100)TITLE1
  100 FORMAT(20A4)
      PRINT 200, TITLE1
  200 FORMAT(1H1,20A4)
      READ(5,130)IRPR,IRMN,IRWS,IRWD
  130 FORMAT(10I6)
      IRSTRT=0
      IF(IRPR.NE.0) IRSTRT=1
      READ(5,110)        NXPOL, XNAME
  110 FORMAT( I6,5A8)
      IF(NXPOL.EQ.0)GO TO 31
      POLNAM(6 )=XNAME
   31 CONTINUE
      READ(5,130) (IPCHOS(I),I=1,6)
      DO 40 I=1,6
      IF(IPCHOS(I).LE.0)  GO TO 41
   40 CONTINUE
   41 IP=I-1
      PRINT 203, (POLNAM(IPCHOS(I)),I=1,IP)
  203 FORMAT(21H0POLLUTANTS SELECTED /6A8)
      READ(5,140)NCASE,(WDSP(I),WSSP(I),I=1,NCASE)
  140 FORMAT(I6,6F6.0)
      IF (NCASE) 48,48,49
   48 PRINT 201
  201 FORMAT(33H0NO SPECIAL WIND CASES CONSIDERED)
      GO TO 51
   49 PRINT 202,        (I,WDSP(I),WSSP(I),I=1,NCASE)
  202 FORMAT(20H0SPECIAL WIND CASES /53H   CASE   WIND DIRECTION(DEGREES
     .)    WIND SPEED(KNOTS)/(I6,F18.2,F23.2))
      DO 50 I=1,NCASE
      WDSP(I)=WDSP(I)* 0.0174533
   50 WSSP(I)=WSSP(I)*0.5148
   51 CONTINUE
      READ(5,120)XBASE,YBASE,INCRX,INCRY,DELTA
  120 FORMAT(2F8.0,2I8,F8.0)                                            MAIN0062
      PRINT 204, XBASE,YBASE,INCRX,INCRY,DELTA                          MAIN0063
  204 FORMAT(43H0LOWER LEFT CORNER OF RECEPTOR GRID IS AT (,F8.3,1H,,   MAIN0064
     .F8.3,1H)/12H   THERE ARE,I4,12H COLUMNS AND,I4,23H ROWS WITH A SPAMAIN0065
     .CING OF,F6.2,11H KILOMETERS)                                      MAIN0066
      NRECEP=0                                                          MAIN0067
      DO 10 I=1,INCRX                                                   MAIN0068
      DO 10 J=1,INCRY                                                   MAIN0069
      NRECEP=NRECEP+1                                                   MAIN0070
      RECEP(1,NRECEP)=XBASE+(I-1)*DELTA                                 MAIN0071
   10 RECEP(2,NRECEP)=YBASE+(J-1)*DELTA                                 MAIN0072
      READ(5,110)IADD                                                   MAIN0073
      IF (IADD) 14,14,15                                                MAIN0074
   14 PRINT 205                                                         MAIN0075
  205 FORMAT(27H0NO SPECIAL RECEPTORS ADDED)                            MAIN0076
      GO TO 21                                                          MAIN0077
   15 PRINT 206                                                         MAIN0078
  206 FORMAT(25H0SPECIAL RECEPTORS ADDED:/36H   NO.   X-COORDINATE   Y-CMAIN0079
     .COORDINATE)                                                       MAIN0080
      DO 20 I=1,IADD                                                    MAIN0081
      READ(5,120)XRECEP,YRECEP                                          MAIN0082
      NRECEP=NRECEP+1                                                   MAIN0083
      PRINT 207, NRECEP,XRECEP,YRECEP                                   MAIN0084
  207 FORMAT(I5,F14.3,F15.3)                                            MAIN0085
      RECEP(1,NRECEP)=XRECEP                                            MAIN0086
      RECEP(2,NRECEP)=YRECEP                                            MAIN0087
   20 CONTINUE                                                          MAIN0088
   21 CONTINUE                                                          MAIN0089
C                                                                       MAIN0090
C     READ AND PRINT STATISTICAL RECEPTOR INPUT                         MAIN0091
C                                                                       MAIN0092
      DO 310 I=1,NRECEP                                                 MAIN0093
      IRSTAT(I)=0                                                       MAIN0094
  310 CONTINUE                                                          MAIN0095
      READ 130,NRSTAT,NSTAPE                                            MAIN0096
      IF (NRSTAT.GT.0) GO TO 305                                        MAIN0097
      PRINT 302                                                         MAIN0098
  302 FORMAT(36H0NO STATISTICAL RECEPTORS CONSIDERED)                   MAIN0099
      IF (NRSTAT.LT.0) PRINT 303                                        MAIN0100
  303 FORMAT(30H0CARDS PUNCHED FOR SYMAP INPUT)                         MAIN0101
      GO TO 400                                                         MAIN0102
  305 PRINT 301,NRSTAT                                                  MAIN0103
  301 FORMAT(1H0,I8,22H STATISTICAL RECEPTORS)                          MAIN0104
      DO 340 I=1,NRSTAT                                                 MAIN0105
      READ 120,XSTARP,YSTARP                                            MAIN0106
      DO 320 IC=1,NRECEP                                                MAIN0107
      IF (XSTARP.EQ.RECEP(1,IC).AND.YSTARP.EQ.RECEP(2,IC)) GO TO 330    MAIN0108
  320 CONTINUE                                                          MAIN0109
      PRINT 321,XSTARP,YSTARP                                           MAIN0110
  321 FORMAT(25H0STATISTICAL RECEPTOR X =,F7.3,5H, Y =,F7.3,            MAIN0111
     . 12H NOT ON GRID)                                                 MAIN0112
      STOP                                                              MAIN0113
  330 IRSTAT(IC)=I                                                      MAIN0114
      PRINT 322,I,XSTARP,YSTARP                                         MAIN0115
  322 FORMAT(I12,7H AT X =,F10.3,4H Y =,F10.3)                          MAIN0116
      RSTAT(1,I)=XSTARP                                                 MAIN0117
      RSTAT(2,I)=YSTARP                                                 MAIN0118
  340 CONTINUE                                                          MAIN0119
  400 CONTINUE                                                          MAIN0120
      CALL  READ                                                        MAIN0121
      CALL MAIN1                                                        MAIN0122
      STOP                                                              MAIN0123
      RETURN                                                            MAIN0124
      END                                                               MAIN0125
```

SUBROUTINE MAINL

Purpose:

To direct the long term calculation by reading the data, calling the wind independent source routine and the diffusion calculation routine.

Input:

1. Card input defining time periods of the calculations.
2. Meteorological data tape.

Output:

1. Common block data to be used by the calculation and output subroutines.
2. A printed summary of the meteorological input data.
3. Statistical receptor data on tape and/or cards for SYMAP.

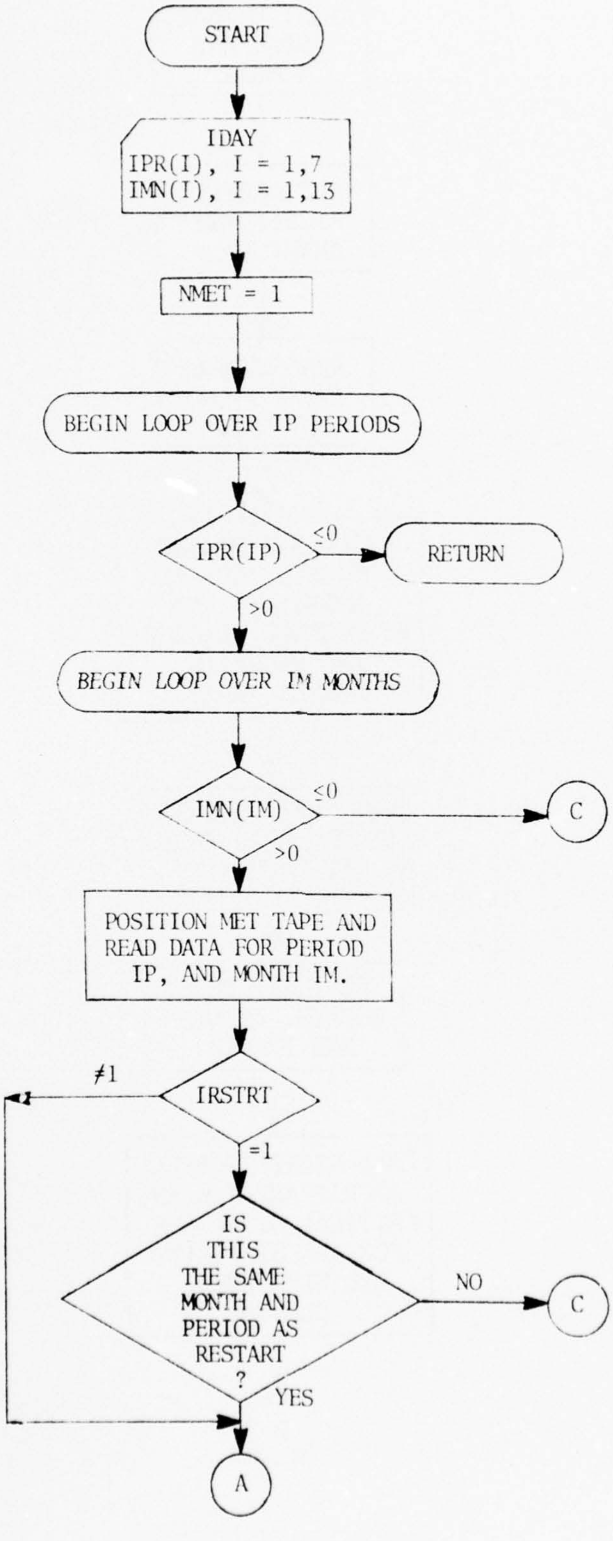
Procedure:

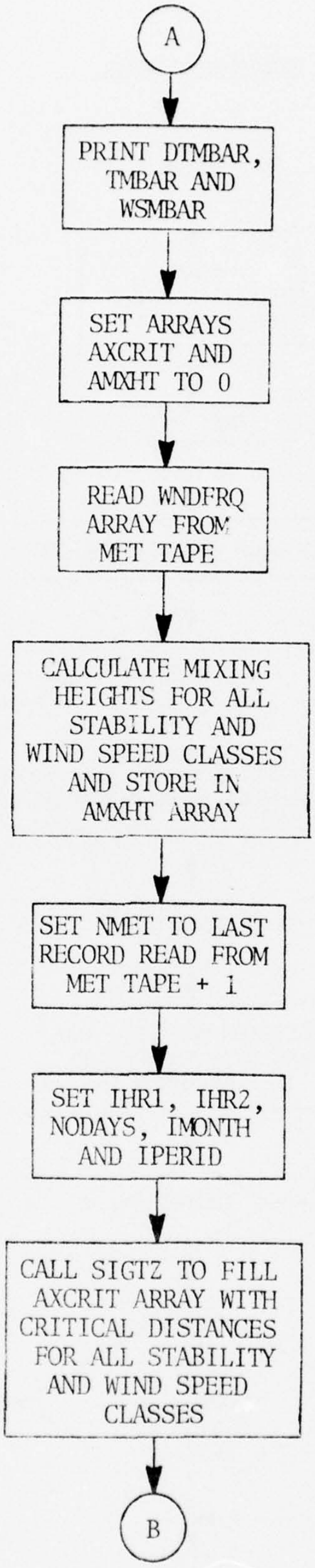
1. Set constants.
2. Read time period data.
3. Read the meteorological data for the month and period selected.
4. Compute mixing heights and critical distances for all stability and wind speed classes.
5. Print input data.
6. Call the wind independent source routines.
7. Call the diffusion model and output routines.
8. Check for statistical output, including cards for SYMAP.

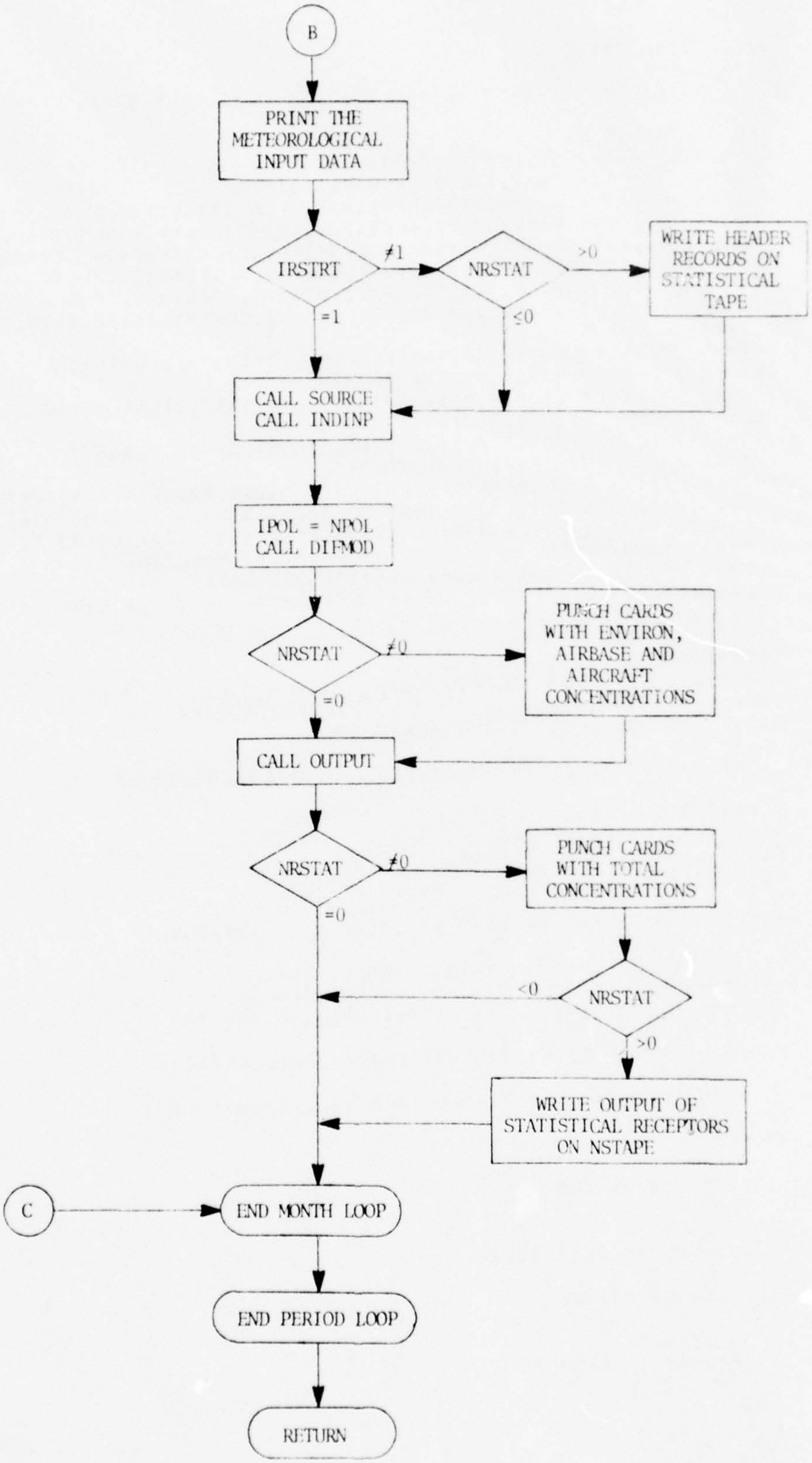
Subroutines Called:

SIGTZ, SOURCE, INDINP, DIFMOD, OUTPUT

SUBROUTINE MAINL


START
IDAY
IPR(I), I = 1,7
IMN(I), I = 1,13
NMET = 1
BEGIN LOOP OVER IP PERIODS
IPR(IP)
≤0
RETURN
>0
BEGIN LOOP OVER IM MONTHS
IMN(IM)
≤0
C
>0
POSITION MET TAPE AND READ DATA FOR PERIOD IP, AND MONTH IM.
≠1
IRSTRT
=1
IS THIS THE SAME MONTH AND PERIOD AS RESTART ?
NO
C
YES
A


A
PRINT DTMBAR, TMBAR AND WSMBAR
SET ARRAYS AXCRIT AND AMXHT TO 0
READ WNDFRQ ARRAY FROM MET TAPE
CALCULATE MIXING HEIGHTS FOR ALL STABILITY AND WIND SPEED CLASSES AND STORE IN AMXHT ARRAY
SET NMET TO LAST RECORD READ FROM MET TAPE + 1
SET IHR1, IHR2, NODAYS, IMONTH AND IPERID
CALL SIGTZ TO FILL AXCRIT ARRAY WITH CRITICAL DISTANCES FOR ALL STABILITY AND WIND SPEED CLASSES
B


B
PRINT THE METEOROLOGICAL INPUT DATA
IRSTRT
≠1
=1
NRSTAT
>0
≤0
WRITE HEADER RECORDS ON STATISTICAL TAPE
CALL SOURCE
CALL INDINP
IPOL = NPOL
CALL DIFMOD
NRSTAT
≠0
=0
PUNCH CARDS WITH ENVIRON, AIRBASE AND AIRCRAFT CONCENTRATIONS
CALL OUTPUT
NRSTAT
≠0
=0
PUNCH CARDS WITH TOTAL CONCENTRATIONS
NRSTAT
<0
>0
WRITE OUTPUT OF STATISTICAL RECEPTORS ON NSTAPE
C
END MONTH LOOP
END PERIOD LOOP
RETURN

```
      SUBROUTINE MAINL
C
C    THIS ROUTINE IS THE MAIN DRIVER FOR THE LONG TERM MODEL
C
      REAL*8 POLNAM
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF
     . ,TEMK,UA
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)
      COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DTMBAR
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /RSTRT/ IPSTRT,IPERID,IRPRX,IRMNX,IRWSX,IRWDX
      COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)
      COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
      COMMON /TITL/ POLNAM( 6),TITLE1(20),IRCHOS( 6),NXPOL,IPX
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
      DIMENSION LHR1(7),LHR2(7),IMN(13),NDM(13),DUM(6),IPR(7)
      DIMENSION SSTAB(6),SWS(6),SWD(16)
      DIMENSION MCAN(4)
      DATA LHR1 /1,7,7,10,16,19,22/,
     .     LHR2 /24,18,9,15,18,21,6/,
     . NDM /31,28,31,30,31,30,31,31,30,31,30,31,365/
      DATA MCAN /4HMONT,4HHLY ,4HANNU,4HAL  /
C
C    READ CARD INPUT DEFINING TIME PERIODS OF CALCULATIONS
C
      READ(5,100)IDAY
      READ(5,100) (IPR(I),I=1,7)
      READ(5,100) (IMN(I),I=1,13)
  100 FORMAT(13I6)
C
C    COMBINE WEEKDAY AND WEEKEND ACTIVITY DISTRIBUTIONS
C
      VHMLDY(1)=VHMLDY(1)*.7142857+VHMLDY(2)*.2857143
      CVABDY(1)=CVABDY(1)*.7142857+CVABDY(2)*.2857143
      CVENDY(1)=CVENDY(1)*.7142857+CVENDY(2)*.2857143
      DO 2 I=1,7
    2 FLDY(1,I)=FLDY(1,I)*.7142857+FLDY(2,I)*.2857143
      DO 3 I=1,8
    3 ACDY(1,I)=ACDY(1,I)*.7142857+ACDY(2,I)*.2857143
      IDAY=1
      NMET=1
C
C    BEGIN LOOP ON PERIODS
C
      DO 10 IP=1,7
      IF(IPR(IP).LE.0)GO TO 20
C
C    BEGIN LOOP ON MONTHS
C
      DO 10 IM=1,13
      IF(IMN(IM).LE.0)GO TO 10
C
```

```
C    POSITION MET TAPE TO READ DATA FOR PERIOD = ID, AND MONTH = IM
C
      IND=(IPR(IP)-1)*13+IMN(IM)
      IF(NMET.EQ.IND) GO TO 11
      INF=IND-1
      DO 12 I=NMET,INF
      IF(MOD(I+12 ,13).EQ.0)READ(10,100)
   12 READ(10,110)D1,D2,D3,D4,D5,D6,D7,WNDFRQ,WNDFRQ
  110 FORMAT(10X,2F6.1,5F10.1,192(/6F10.0))
   11 IF(MOD(IND+12,13).EQ.0)READ(10,100)
      READ(10,140)TMBAR,DTMBAR,D1,P,D2,VSMBAR,PCTCWB
  140 FORMAT(10X,2F6.1,5F10.1)
      IF(IRSTRT.NE.1) GO TO 40
C
C    IF RESTARTING, CHECK THE PERIOD AND MONTH AGAINST INPUT
C
      IF(IPR(IP).NE.IRPRX) GO TO 10
      IF(IMN(IM).NE.IRMNX) GO TO 10
   40 CONTINUE
C
C    PRINT MONTHLY OR ANNUAL DATA
C
      NAOM=0
      IF (IMN(IM).EQ.13) NAOM=1
      WRITE (6,130) DTMBAR,(MOAN(NAOM*2+I),I=1,2),TMBAR,
     . (MOAN(NAOM*2+I),I=1,2),VSMBAR
  130 FORMAT(24HOTEMPERATURE VARIATION =,F10.1/
     . 9H AVERAGE ,2A4,13HTEMPERATURE =,F10.1/
     . 9H AVERAGE ,2A4,12HWIND SPEED =,F10.1)
C
C    READ THE WIND FREQUENCY DATA AND CALCULATE MIXING HEIGHTS
C    FOR ALL STABILITY AND WIND SPEED CLASSES
C
      DO 13 I=1,6
      DO 14 K=1,6
      AXCRIT(K,I)=0.0
      AMXHT(K,I)=0.0
   14 CONTINUE
      READ(10,120)((WNDFRQ(K,J,I),K=1,6),J=1,16)
  120 FORMAT(6F10.0)
      DO 17 J=1,16
      READ(10,120) (DUM(K),K=1,6)
      DO 15 K=1,6
      IF(WNDFRQ(K,J,I).EQ.0.0)GO TO 15
      AMXHT(K,I)=AMXHT(K,I)+DUM(K)   *WNDFRQ(K,J,I)
      AXCRIT(K,I)=AXCRIT(K,I)+WNDFRQ(K,J,I)
   15 CONTINUE
   17 CONTINUE
      DO 16 K=1,6
      IF (AXCRIT(K,I).EQ.0.0) GO TO 16
      AMXHT(K,I)=AMXHT(K,I)/AXCRIT(K,I)
   16 CONTINUE
   13 CONTINUE
      NMET=IND+1
C
C    SET HOURS, MONTH, PERIOD AND NUMBER OF DAYS
C
      IHR1=IHR1(IPR(IP))
      IHR2=IHR2(IPR(IP))
      NODAYS=NDM(IMN(IM))
      IMONTH=IMN(IM)
      IPERID=IPR(IP)
```

```
C
C   CALL SIGTZ TO FILL THE AXCRIT ARRAY WITH CRITICAL DISTANCES
C   FOR ALL STABILITY AND WIND SPEED CLASSES
C
      KER=0
      CALL SIGTZ(II,D,DD)
C
C   PRINT MET INPUT DATA
C
      PA=PA/100.
      TEMF=TMBAR
      TEMK=(TEMF-32.)*5./9.+273.
      DO 4193 L=1,6
      SSTAB(L)=0.0
 4193 SWS(L)=0.0
      DO 4194 I=1,16
 4194 SWD(I)=0.0
      DO 4190 L=1,6
      WRITE(6,4095)IMONTH,IHR1,IHR2
 4095 FORMAT(1H1,T40,34HMETEOROLOGICAL INPUT DATA -- MONTH,I3,7H, HOURS,
     . I3,4H00 -,I3,2H00/1H0)
      WRITE(6,5017)TEMK,P
 5017 FORMAT(T30,21HAMBIENT TEMPERATURE =,F7.1,2H K/
     . T30,18HAMBIENT PRESSURE =,F6.0,10H MILLIBARS)
      WRITE(6,5100)L
 5100 FORMAT(1H0,T30,15HSTABILITY CLASS,I2/T69,16HWIND SPEED CLASS/
     . 1H0,T32,14HWIND DIRECTION,8X,1H1,8X,1H2,8X,1H3,8X,1H4,8X,1H5,8X,
     . 1H6)
      DO 4191 K=1,16
      DO 4192 J=1,6
      SSTAB(L)=SSTAB(L)+WNDFRQ(J,K,L)
      SWS(J)=SWS(J)+WNDFRQ(J,K,L)
 4192 SWD(K)=SWD(K)+WNDFRQ(J,K,L)
 4191 WRITE(6,5105) K,(WNDFRQ(J,K,L),J=1,6)
 5105 FORMAT(1H0,T38,I2,9X,6F9.4)
      WRITE(6,4090) (AMXHT(K,L),K=1,6)
 4090 FORMAT(/1H0,T30,14HMIXING DEPTH =,3X,6F9.0,7H METERS)
 4190 CONTINUE
      WRITE(6,4195)IMONTH,IHR1,IHR2
 4195 FORMAT(1H1,T30,40HSUMMARY OF METEOROLOGICAL DATA -- MONTH,I3,
     . 7H, HOURS,I3,4H00 -,I3,2H00/1H0,T52,22HFREQUENCY OF OCCURANCE)
      AVSTAB=0.0
      AVWS=0.0
      DO 4200 J=1,6
      AJ=J
      AVSTAB=AVSTAB+AJ*SSTAB(J)
 4200 AVWS=AVWS+UU(J)*SWS(J)
      WRITE(6,4196) (J,SSTAB(J),J=1,6)
 4196 FORMAT(1H0,T31,15HSTABILITY CLASS,6(/T40,I6,F12.4))
      WRITE(6,4201) AVSTAB
 4201 FORMAT(T61,17HAVERAGE STABILITY,F6.2)
      WRITE(6,4197) (J,SWS(J),J=1,6)
 4197 FORMAT(1H0,T30,16HWIND SPEED CLASS,6(/T40,I6,F12.4))
      WRITE(6,4202) AVWS
 4202 FORMAT(T61,18HAVERAGE WIND SPEED,F8.3)
      WRITE(6,4198) (J,SWD(J),J=1,16)
 4198 FORMAT(1H0,T32,14HWIND DIRECTION,16(/T40,I6,F12.4))
      SWD1=0.5*SWD(1)+SWD(2)+SWD(3)+SWD(4)+0.5*SWD(5)
      SWD2=0.5*SWD(5)+SWD(6)+SWD(7)+SWD(8)+0.5*SWD(9)
      SWD3=0.5*SWD(9)+SWD(10)+SWD(11)+SWD(12)+0.5*SWD(13)
      SWD4=0.5*SWD(13)+SWD(14)+SWD(15)+SWD(16)+0.5*SWD(1)
      WRITE(6,4199)SWD1,SWD2,SWD3,SWD4
 4199 FORMAT(1H0,T33,13HWIND QUADRANT/T44,2HNE,F12.4/T44,2HSE,F12.4/
     . T44,2HSW,F12.4/T44,2HNW,F12.4/1H1)
      IF (IRSTRT.EQ.1) GO TO 50
C
C   IF NOT RESTARTING, CHECK FOR THE STATISTICAL OPTION AND
C   WRITE HEADER RECORDS ON NSTAPE
C
      IF (NRSTAT.GT.0) WRITE (NSTAPE) IPERID,IMONTH,NODAYS,IDAY,IHR1,
     . IHR2,NPOL,(POLNAM(I),I=1,NPOL),NRSTAT,((RSTAT(I,J),I=1,2),J=1,
     . NRSTAT),TITLE1
      IF (NRSTAT.GT.0) WRITE (NSTAPE) WNDFRQ
   50 CONTINUE
C
C   CALL THE NON-AIRCRAFT SOURCE ROUTINES AND PRINT THE
C   WIND INDEPENDENT INPUT
C
      CALL SOURCE
      CALL INDINP
C
C   CALL THE DIFFUSION MODEL
C
      IPOL=NPOL
      CALL DIFMOD
      IF (NRSTAT.EQ.0) GO TO 60
C
C   IF SYMAP OPTION IS CHOSEN, PUNCH CARDS FOR
C   ENVIRON, AIRBASE AND AIRCRAFT CONCENTRATIONS
C
      DO 59 K=1,3
      PUNCH 57,NRECEP,K
   57 FORMAT(2I6)
      DO 59 N=1,NRECEP
      PUNCH 58,(RECEP(I,N),I=1,2),(RECDAT(K,J,N),J=1,NPOL)
   58 FORMAT(1P8E10.3)
   59 CONTINUE
   60 CONTINUE
C
C   PRINT RESULTS
C
      CALL OUTPUT
      IF (NRSTAT.EQ.0) GO TO 18
C
C   IF SYMAP OPTION IS CHOSEN PUNCH TOTAL CONCENTRATIONS
C
      K=4
      PUNCH 57,NRECEP,K
      DO 19 N=1,NRECEP
      PUNCH 58,(RECEP(I,N),I=1,2),(RECDAT(1,J,N),J=1,NPOL)
   19 CONTINUE
      IF (NRSTAT.LT.0) GO TO 18
C
C   IF STATISTICAL OPTION IS CHOSEN RECORD THE OUTPUT
C
      WRITE (NSTAPE) NRSTAT
      DO 49 N=1,NRSTAT
      K=IRSTAT(N)
      IF (K.NE.0) WRITE (NSTAPE) (RECDAT(1,J,K),J=1,NPOL)
   49 CONTINUE
   18 CONTINUE
   10 CONTINUE
   20 CONTINUE
      RETURN
      END                                                               MAINL248
```

SUBROUTINE METHA

Purpose:

To calculate diurnal emissions allowing each source in a class to have a unique or default distribution pattern.

Input:

1. The ICLASS number of the sources and NPTC, the number of sources not using the default of a uniform distribution.
2. For each of the NPTC sources, the source ID number and fractions of the hour, day and month, FH, FD and FM, which that source is on. If one or two of the fractions are left blank, the default is used. If all are blank, the source is assumed to be off.

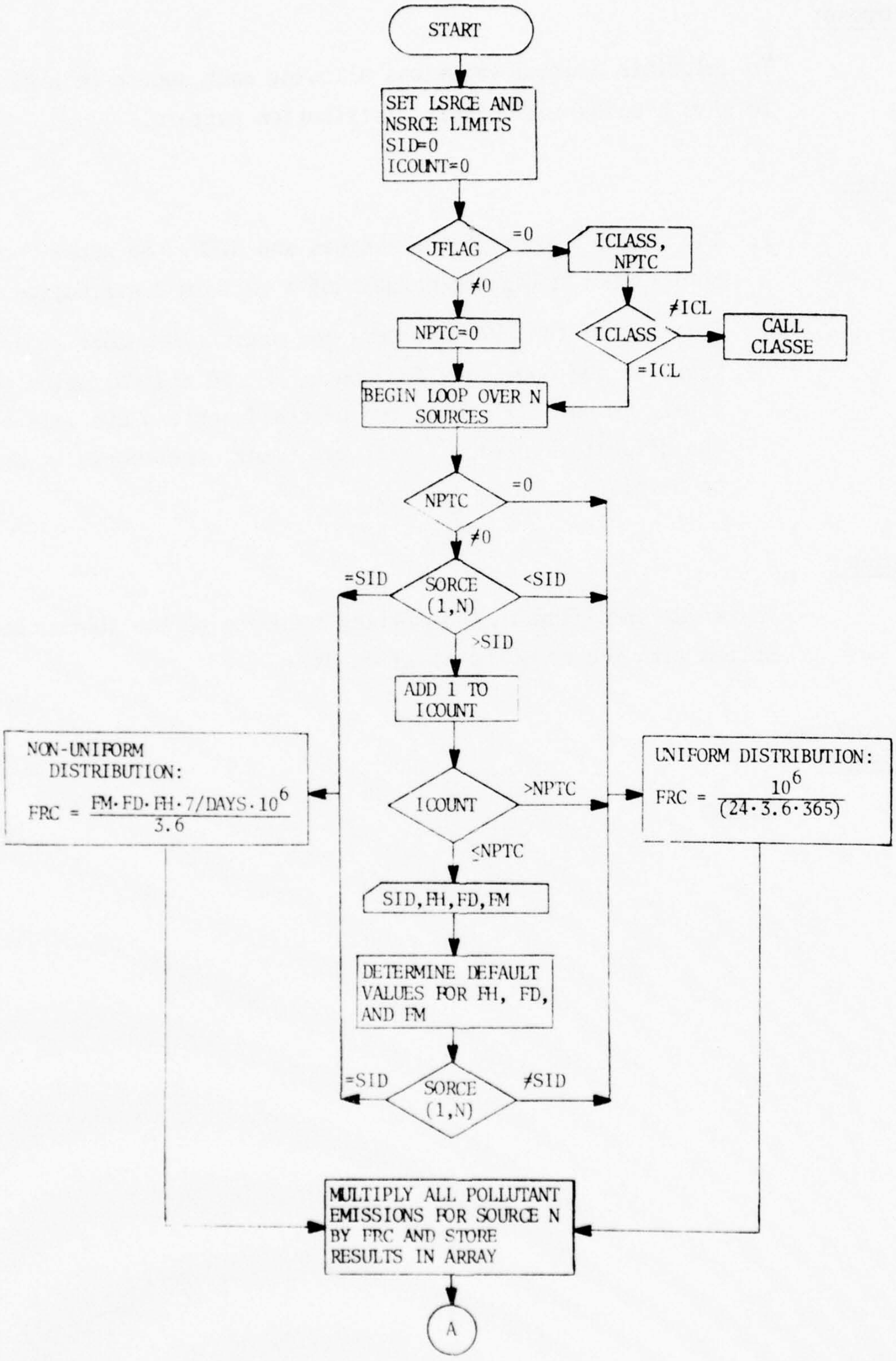
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

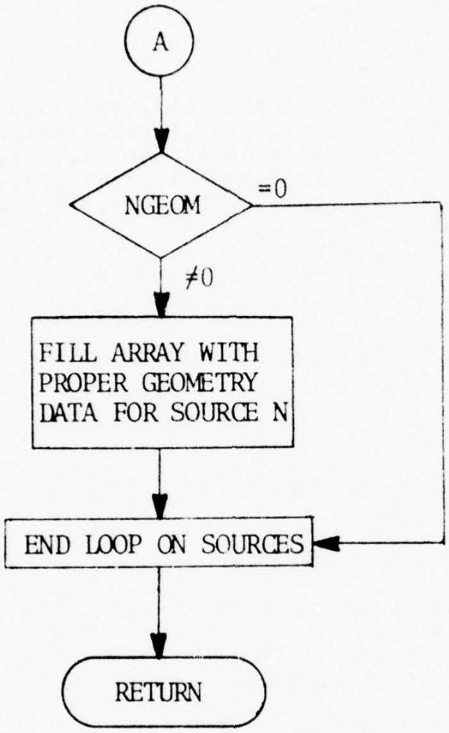
Subroutines Called:

CLASSE

SUBROUTINE METHA


START
SET LSRCE AND NSRCE LIMITS
SID=0
ICOUNT=0
JFLAG
=0
≠0
ICLASS, NPTC
ICLASS
≠ICL
=ICL
CALL CLASSE
NPTC=0
BEGIN LOOP OVER N SOURCES
NPTC
=0
≠0
SORCE (1,N)
=SID
<SID
>SID
ADD 1 TO ICOUNT
NON-UNIFORM DISTRIBUTION:
FRC = FM·FD·FH·7/DAYS·10^6 / 3.6
ICOUNT
>NPTC
≤NPTC
UNIFORM DISTRIBUTION:
FRC = 10^6 / (24·3.6·365)
SID,FH,FD,FM
DETERMINE DEFAULT VALUES FOR FH, FD, AND FM
SORCE (1,N)
=SID
≠SID
MULTIPLY ALL POLLUTANT EMISSIONS FOR SOURCE N BY FRC AND STORE RESULTS IN ARRAY
A

SUBROUTINE METHA (Contd.)



```
      SUBROUTINE METHA(MAXN,ARRAY,I1,I2,ICL)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS ALLOWING EACH
C     SOURCE IN A CLASS TO HAVE A DIFFERENT DISTRIBUTION PATTERN.
C     DEFAULTS ARE: FH = 1/24
C                   FD = 1/7
C                   FM = 1/12 OR 1
C
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                  ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      SID=0.
      ICOUNT=0
      IF (JFLAG.EQ.0) GO TO 5
      NPTC=0
      GO TO 6
    5 READ 1,ICLASS,NPTC
    1 FORMAT(2I4)
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
    6 DO 100 N=LSRCE,NSRCE
      IF (NPTC.EQ.0) GO TO 30
      IF (SID-SORCE(1,N))10,40,30
 10   ICOUNT=ICOUNT+1
      IF(ICOUNT.GT.NPTC) GO TO 30
      READ 2,SID,FH,FD,FM
    2 FORMAT(F4.0,4X,3F8.7)
      IF (FH+FD+FM.EQ.0.0) GO TO 20
C
C     DETERMINE DEFAULT VALUES
C
      IF (FM.NE.0.0) GO TO 15
      FM=1./12.
      IF (DAYS.GE.365.) FM=1.
   15 IF (FD.EQ.0.0) FD=1./7.
      IF (FH.EQ.0.0) FH=1./24.
C
   20 CONTINUE
      IF (SID-SORCE(1,N))30,40,30
C
C     UNIFORM DISTRIBUTION
C
   30 FRC=1.0E+6/(24.*3.6*365.)
      GO TO 50
C
C     NON-UNIFORM DISTRIBUTION
C
   40 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)
   50 DO 60 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   60 CONTINUE
      IF (NGEOM.EQ.0) GO TO 100
      DO 70 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   70 CONTINUE
      IF (IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
  100 CONTINUE
```

```
      RETURN                                                            METHA062
      END                                                               METHA063
```

SUBROUTINE METHB

Purpose:

To calculate diurnal emissions using a degree-hour method.

Input:

The ICLASS number of the sources and UNIFRC, the fraction of emissions which are to be uniformly distributed.

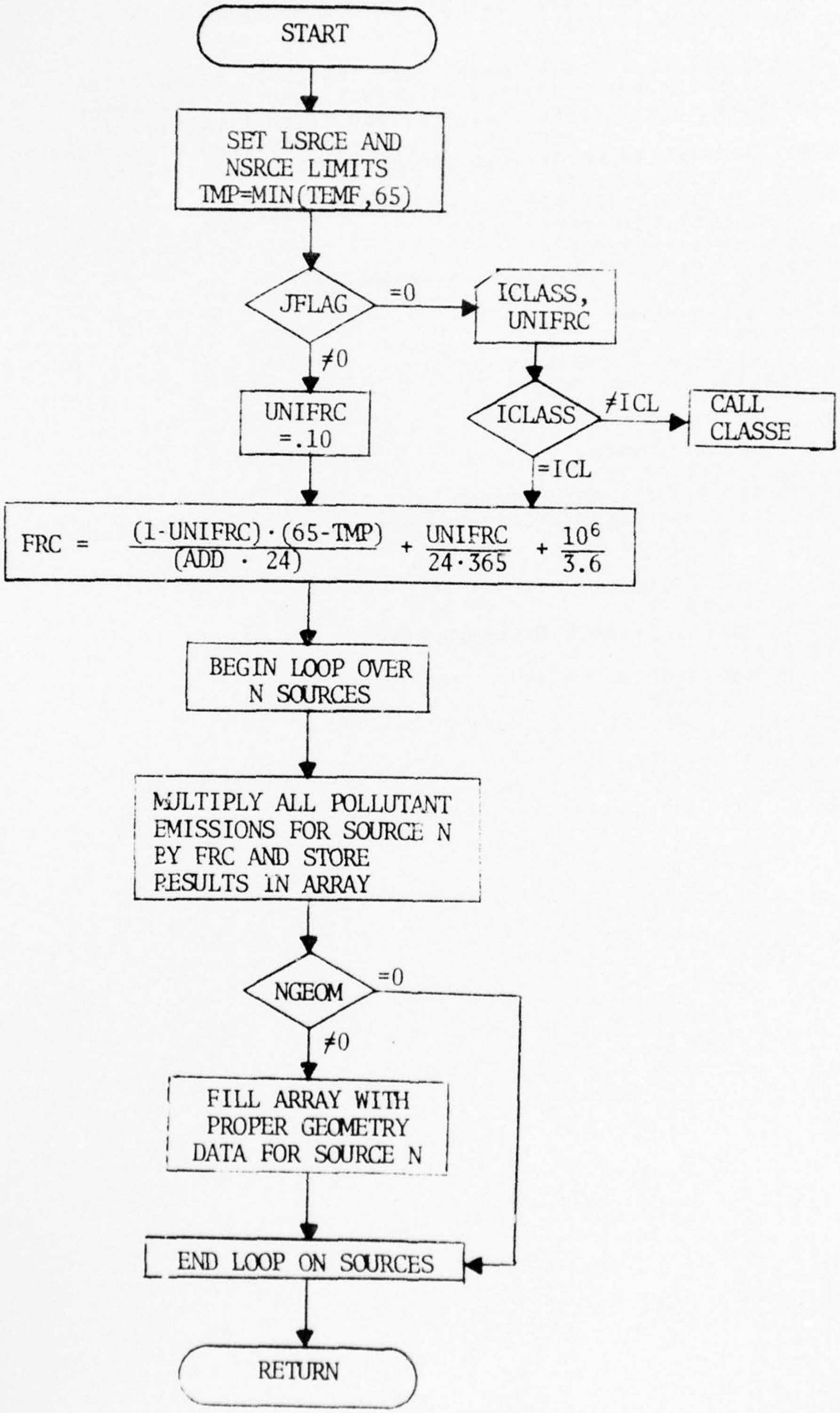
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

CLASSE

SUBROUTINE METHB


START
SET LSRCE AND NSRCE LIMITS TMP=MIN(TEMF,65)
JFLAG
=0
≠0
ICLASS, UNIFRC
UNIFRC =.10
ICLASS
≠ICL
=ICL
CALL CLASSE
$FRC = \frac{(1-UNIFRC)\cdot(65-TMP)}{(ADD \cdot 24)} + \frac{UNIFRC}{24\cdot 365} + \frac{10^6}{3.6}$
BEGIN LOOP OVER N SOURCES
MULTIPLY ALL POLLUTANT EMISSIONS FOR SOURCE N BY FRC AND STORE RESULTS IN ARRAY
NGEOM
=0
≠0
FILL ARRAY WITH PROPER GEOMETRY DATA FOR SOURCE N
END LOOP ON SOURCES
RETURN

```
      SUBROUTINE METHB(MAXN,ARRAY,I1,I2,ICL)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING
C     A DEGREE-HOUR METHOD
C
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                      ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,
     .  JSTAB,HLID,TEMF,TEMK
      DIMENSION ARRAY(I1,I2)
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      TMP=TEMF
      IF (TEMF.GT.65.) TMP=65.
      IF (JFLAG.EQ.0) GO TO 5
      UNIFRC=.10
      GO TO 6
    5 READ 1,ICLASS,UNIFRC
    1 FORMAT(I4,4X,F8.7)
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
    6 FRC=(((1.0-UNIFRC)*((65.0-TMP)/(ADD*24.0)))+(UNIFRC/(24.0*365.0)))
     .    * (1.0E+6/3.6)
C
      DO 30 N=LSRCE,NSRCE
      DO 10 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   10 CONTINUE
      IF (NGEOM.EQ.0) GO TO 30
      DO 20 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
```

SUBROUTINE METHC

Purpose:

To calculate diurnal emissions using the same distribution pattern for all sources in the class.

Input:

The ICLASS number of the sources and the fractions of the hour, day and month, FH, FD and FM, which the sources are on. If one or two of the fractions are left blank, the default is used. If all are blank, the sources are assumed to be off.

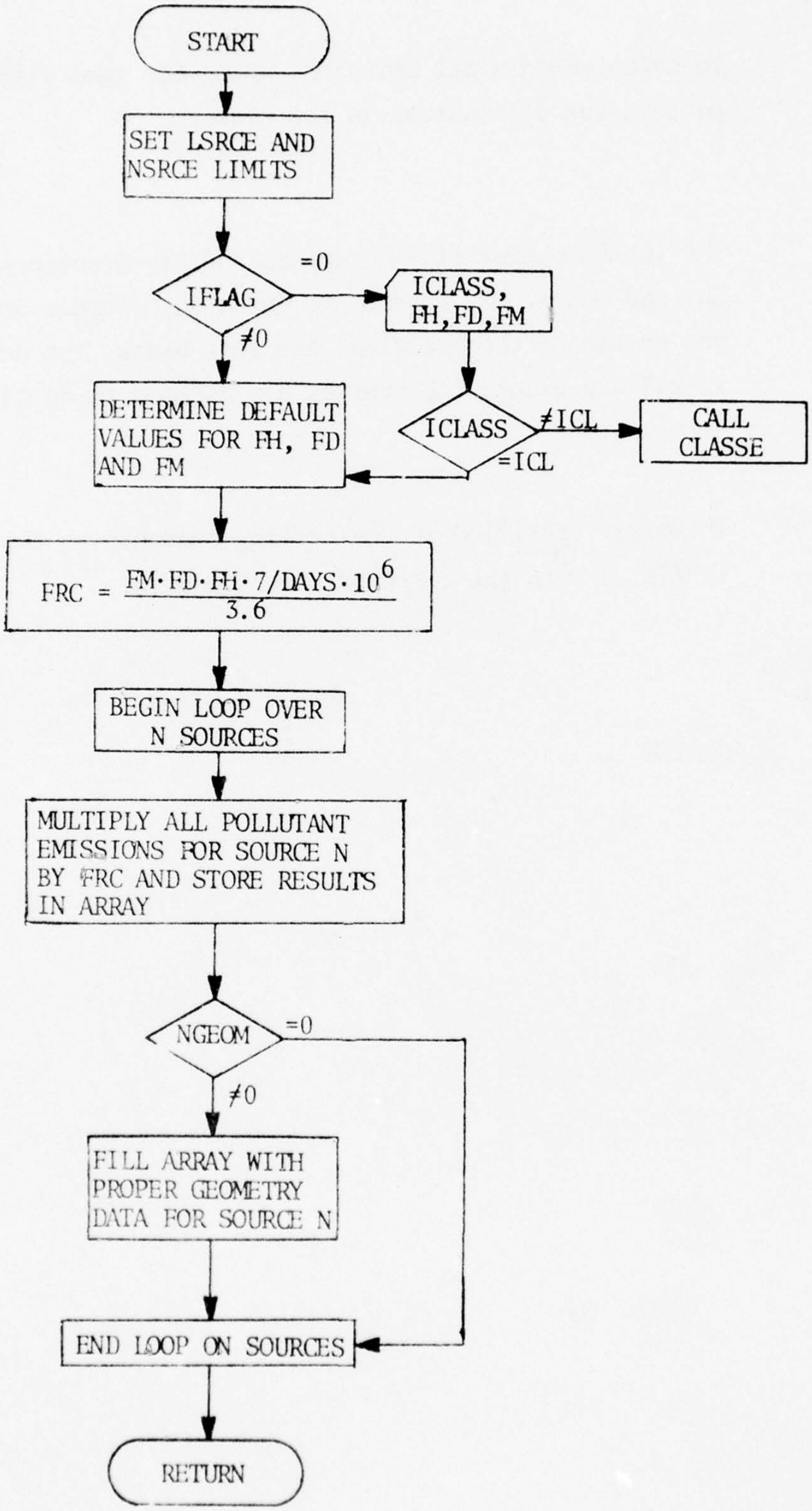
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

CLASSE

SUBROUTINE METHC


START
SET LSRCE AND NSRCE LIMITS
IFLAG
=0
≠0
ICLASS, FH,FD,FM
ICLASS
≠ICL
=ICL
CALL CLASSE
DETERMINE DEFAULT VALUES FOR FH, FD AND FM
FRC = $\frac{FM \cdot FD \cdot FH \cdot 7/DAYS \cdot 10^6}{3.6}$
BEGIN LOOP OVER N SOURCES
MULTIPLY ALL POLLUTANT EMISSIONS FOR SOURCE N BY FRC AND STORE RESULTS IN ARRAY
NGEOM
=0
≠0
FILL ARRAY WITH PROPER GEOMETRY DATA FOR SOURCE N
END LOOP ON SOURCES
RETURN

```
      SUBROUTINE METHC(MAXN,ARRAY,I1,I2,ICL)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE SAME
C     DISTRIBUTION PATTERN FOR ALL SOURCES IN A CLASS.
C     DEFAULTS ARE: FH = 0 OR 1/12 OR 1/24
C                   FD = 1/7
C                   FM = 1/12 OR 1
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                   ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      IF (JFLAG.EQ.0) GO TO 5
      FD=1./7.
      FM=0.0
      FH=0.0
      GO TO 6
    5 READ 1,ICLASS,FH,FD,FM
    1 FORMAT(I4,4X,3F8.7)
      IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)
      IF (FH+FD+FM.EQ.0.0) GO TO 10
C
C     DETERMINE DEFAULT VALUES
C
      IF (FD.EQ.0.0) FD=1./7.
    6 IF (FM.NE.0.0) GO TO 7
      FM=1./12.
      IF (DAYS.GE.365.) FM=1.
    7 IF (FH.NE.0.0) GO TO 10
      IF(IHR1.GT.6.AND.IHR1.LT.19.AND.IHR2.GT.6.AND.IHR2.LT.19)FH=1./12.
      IF (IHR1.EQ.1.AND.IHR2.EQ.24) FH=1./24.
   10 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)
C
   20 DO 100 N=LSRCE,NSRCE
      DO 30 I=1,NPLTS
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC
   30 CONTINUE
      IF (NGEOM.EQ.0) GO TO 100
      DO 40 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   40 CONTINUE
      IF (IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
  100 CONTINUE
      RETURN
      END
```

SUBROUTINE METHD

Purpose:

To calculate diurnal emissions using the temporal distribution arrays for fuel handling activities.

Input:

None

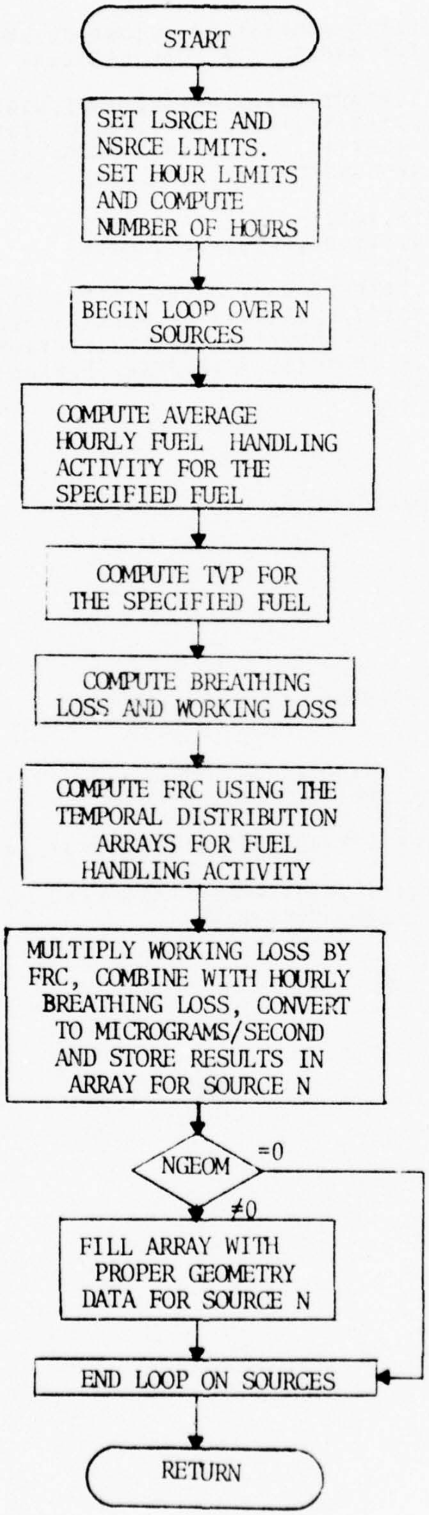
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

None

SUBROUTINE METHD


START
SET LSRCE AND NSRCE LIMITS. SET HOUR LIMITS AND COMPUTE NUMBER OF HOURS
BEGIN LOOP OVER N SOURCES
COMPUTE AVERAGE HOURLY FUEL HANDLING ACTIVITY FOR THE SPECIFIED FUEL
COMPUTE TVP FOR THE SPECIFIED FUEL
COMPUTE BREATHING LOSS AND WORKING LOSS
COMPUTE FRC USING THE TEMPORAL DISTRIBUTION ARRAYS FOR FUEL HANDLING ACTIVITY
MULTIPLY WORKING LOSS BY FRC, COMBINE WITH HOURLY BREATHING LOSS, CONVERT TO MICROGRAMS/SECOND AND STORE RESULTS IN ARRAY FOR SOURCE N
NGEOM
=0
≠0
FILL ARRAY WITH PROPER GEOMETRY DATA FOR SOURCE N
END LOOP ON SOURCES
RETURN

```
      SUBROUTINE METHD(MAXN,ARRAY,I1,I2)
C
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE
C     TEMPORAL DISTRIBUTION ARRAYS FOR FUEL HANDLING ACTIVITIES
C
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),
     .                    ABPT(16,150),ABAR(11,100),ABLN(14,100)
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      COMMON/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2
      COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,
     .  JSTAB,HLID,TEMF,TEMK
      COMMON / DEFALT /  ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
      COMMON /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
      COMMON/MONMET/ TMBAR
      DIMENSION ARRAY(I1,I2)
      LSRCE=NSRCE+1
      NSRCE=NSRCE+MAXN
      NHI=IHR2
      IF(IHR1.GT.IHR2) NHI=24+IHR2
      HRS=NHI-IHR1 + 1
      DO 30 N=LSRCE,NSRCE
      FLHOUR=0.
      IDF=SORCE(14,N)
      DO 10 I=IHR1,NHI
      II=I
      IF(I.GT.24) II=I-24
      FLHOUR=FLHR(II,IDF)+FLHOUR
   10 CONTINUE
      FLHOUR=FLHOUR/HRS
      TVP=EXP(ALPHA(IDF)-BETA(IDF)/(5.*(TMBAR-32.)/9.+273.))
      BRLOSS=SORCE(13,N)*(TVP/(14.7-TVP))**0.69
      WRKLOS=SORCE(12,N)*TVP
      FRC=FLMO(IMONTH,IDF)*FLDY(IDAY,IDF)*FLHOUR*(7./DAYS)
C
      ARRAY(12,N)=(BRLOSS/(365.*24.)+WRKLOS*FRC)*1.E+6/3.6
      IF(IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)
      IF (NGEOM.EQ.0) GO TO 30
      DO 20 I=1,NGEOM
      ARRAY(I,N)=SORCE(I+2,N)
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
```

SUBROUTINE METHE

Purpose:

To calculate diurnal emissions using the temporal distribution arrays for vehicle activities.

Input:

None

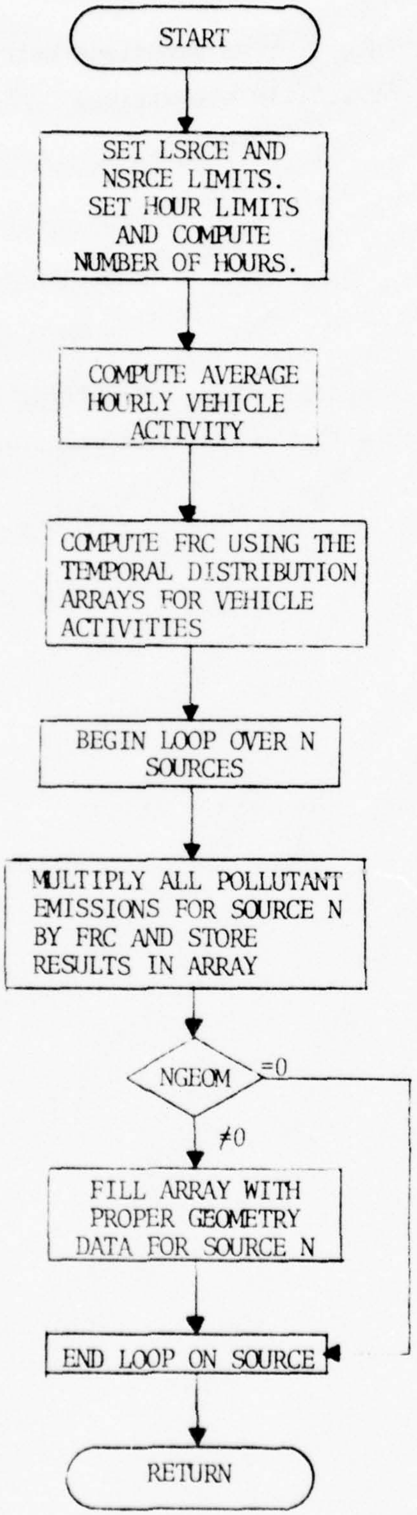
Output:

The array specified in the calling sequence to the subroutine is filled with the computed emission data.

Subroutines Called:

None

SUBROUTINE METHE


START
SET LSRCE AND NSRCE LIMITS. SET HOUR LIMITS AND COMPUTE NUMBER OF HOURS.
COMPUTE AVERAGE HOURLY VEHICLE ACTIVITY
COMPUTE FRC USING THE TEMPORAL DISTRIBUTION ARRAYS FOR VEHICLE ACTIVITIES
BEGIN LOOP OVER N SOURCES
MULTIPLY ALL POLLUTANT EMISSIONS FOR SOURCE N BY FRC AND STORE RESULTS IN ARRAY
NGEOM
=0
≠0
FILL ARRAY WITH PROPER GEOMETRY DATA FOR SOURCE N
END LOOP ON SOURCE
RETURN

```
      SUBROUTINE METHE(MAXN,ARRAY,ARMO,ARDY,ARHR,I1,I2)                 METHE000
C                                                                       METHE001
C     THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE               METHE002
C     TEMPORAL DISTRIBUTION ARRAYS FOR VEHICLE ACTIVITES                METHE003
C                                                                       METHE004
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,          METHE005
     . NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),         METHE006
     .                  ABPT(16,150),ABAR(11,100),ABLN(14,100)          METHE007
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)          METHE008
     . ,LOC1,LOC2,NGEOM,IPT                                             METHE009
      COMMON/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2                        METHE010
      DIMENSION ARMO(13),ARDY(2),ARHR(24),ARRAY(I1,I2)                  METHE011
      LSRCE=NSRCE+1                                                     METHE012
      NSRCE=NSRCE+MAXN                                                  METHE013
      ARHOUR=0.                                                         METHE014
      NHI=IHR2                                                          METHE015
      IF(IHR1.GT.IHR2) NHI=24+IHR2                                      METHE016
      HRS=NHI-IHR1+1                                                    METHE017
      DO 10 I=IHR1,NHI                                                  METHE018
      II=I                                                              METHE019
      IF(I.GT.24) II=I-24                                               METHE020
      ARHOUR=ARHOUR+ARHR(II)                                            METHE021
   10 CONTINUE                                                          METHE022
      ARHOUR=ARHOUR/HRS                                                 METHE023
      FRC=ARMO(IMONTH)*ARDY(IDAY)*ARHOUR*(7./DAYS)*(1E+6/3.6)           METHE024
C                                                                       METHE025
      DO 40 N=LSRCE,NSRCE                                               METHE026
      DO 20 I=1,NPLTS                                                   METHE027
      ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC                               METHE028
   20 CONTINUE                                                          METHE029
      IF (NGEOM.EQ.0) GO TO 40                                          METHE030
      DO 30 I=1,NGEOM                                                   METHE031
   30 ARRAY(I,N)=SORCE(I+2,N)                                           METHE032
   40 CONTINUE                                                          METHE033
      RETURN                                                            METHE034
      END                                                               METHE035
```

SUBROUTINE OUTPUT

Purpose:

To print the pollutant concentrations at all receptors for the environ, airbase, aircraft and total combined sources.

Input:

1. Title information.
2. The RECEP and RECDAT arrays containing receptor and concentration data.

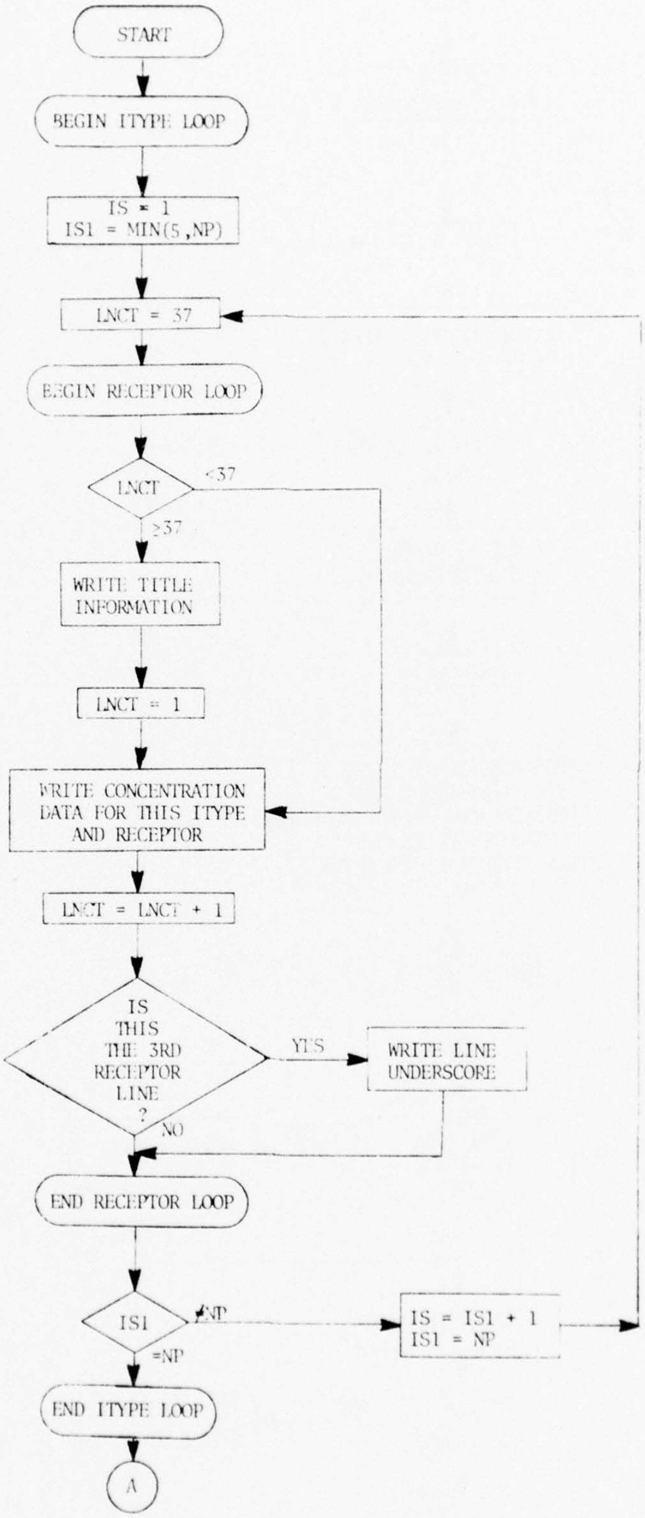
Output:

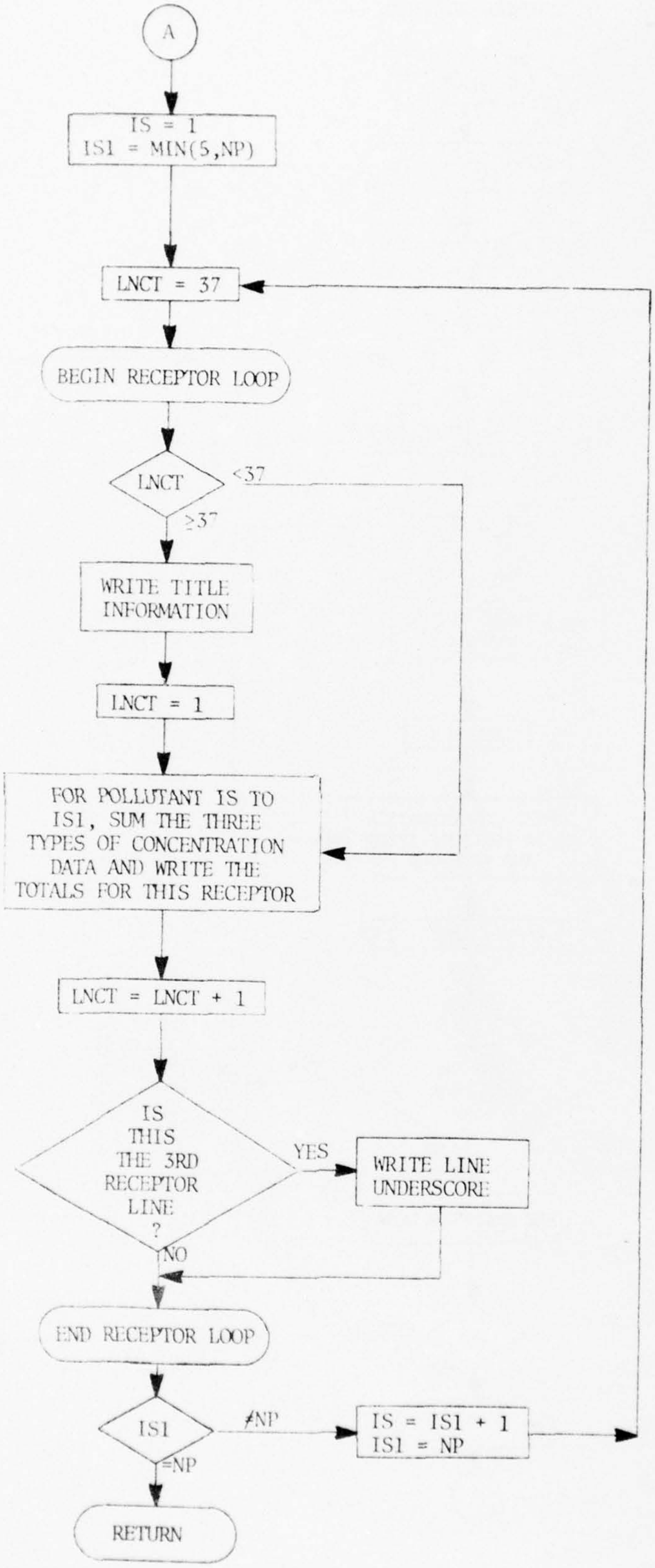
Printed concentration data.

Subroutines Called:

None

SUBROUTINE OUTPUT




A
IS = 1
IS1 = MIN(5,NP)
LNCT = 37
BEGIN RECEPTOR LOOP
LNCT
<37
≥37
WRITE TITLE INFORMATION
LNCT = 1
FOR POLLUTANT IS TO IS1, SUM THE THREE TYPES OF CONCENTRATION DATA AND WRITE THE TOTALS FOR THIS RECEPTOR
LNCT = LNCT + 1
IS THIS THE 3RD RECEPTOR LINE ?
YES
WRITE LINE UNDERSCORE
NO
END RECEPTOR LOOP
IS1
≠NP
IS = IS1 + 1
IS1 = NP
=NP
RETURN

```
      SUBROUTINE OUTPUT
C
C    THIS ROUTINE PRINTS THE POLLUTANT CONCENTRATION AT ALL
C    RECEPTORS FOR THE ENVIRON, AIRBASE, AIRCRAFT AND TOTAL
C    COMBINED SOURCES.
C
      REAL*8 POLNAM
      REAL*8     SORNAM(4)
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,NP
      DIMENSION NNM(13),NNHR(25),NND(2)
      DATA SORNAM/7HENVIRON,7HAIRPORT,8HAIRCRAFT,5HTOTAL /
      DATA NNHR/4H0000,4H0100,4H0200,4H0300,4H0400,4H0500,4H0600,
     .4H0700,4H0800,4H0900,4H1000,4H1100,4H1200,4H1300,4H1400,4H1500,
     .4H1600,4H1700,4H1800,4H1900,4H2000,4H2100,4H2200,4H2300,4H2400/,
     . NND /4HDAY ,4HEND /,
     .      NNM/4HJAN ,4HFEB ,4HMAR ,4HAPR ,4HMAY ,4HJUN ,4HJUL ,
     .         4HAUG ,4HSEP ,4HOCT ,4HNOV ,4HDEC ,4HYEAR/
      DO 100 ITYPE=1,3
      IS=1
      IS1=MIN0(5,NP)
  110 LNCT=37
      DO 120 IRECEP=1,NRECEP
      IF (LNCT.LT.37) GO TO 121
      WRITE(6,220)TITLE1,NNM(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
      WRITE(6,200)SORNAM (ITYPE),(POLNAM(IPCHOS(J)),J=IS,IS1)
      WRITE (6,260)
      LNCT=1
  121 WRITE(6,210)IRECEP,(RECEP(J,IRECEP),J=1,2),
     .(RECDAT(ITYPE,IPCHOS(K ),IRECEP),K =IS,IS1)
      LNCT=LNCT+1
      IF(MOD(IRECEP,3).EQ.0)WRITE(6,260)
  120 CONTINUE
      IF (IS1.EQ.NP) GO TO 100
      IS=IS1+1
      IS1=NP
      GO TO 110
  100 CONTINUE
      IS=1
      IS1=MIN0(5,NP)
  125 LNCT=37
      DO 130 IRECEP=1,NRECEP
      IF (LNCT.LT.37) GO TO 133
      WRITE(6,220)TITLE1,NNM(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
      WRITE(6,200)SORNAM (   4   ),(POLNAM(IPCHOS(J)),J=IS,IS1)
      WRITE (6,260)
      LNCT=1
  133 CONTINUE
      DO 131 J=IS,IS1
      DO 131 K=2,3
  131 RECDAT(1,IPCHOS(J),IRECEP)=RECDAT(1,IPCHOS(J),IRECEP)+
     . RECDAT(K,IPCHOS(J),IRECEP)
      WRITE(6,210)IRECEP,(RECEP(J,IRECEP),J=1,2),
     . (RECDAT(1,IPCHOS(K ),IRECEP),K=IS,IS1)
      LNCT=LNCT+1
      IF(MOD(IRECEP,3).EQ.0)WRITE(6,260)
  130 CONTINUE
      IF (IS1.EQ.NP) GO TO 140
      IS=IS1+1
      IS1=NP
      GO TO 125                                                         OUTPT062
 140  CONTINUE                                                          OUTPT063
 200  FORMAT(1H0,96(1H-)/2H I,22X,33HRECEPTOR CONCENTRATION DATA FROM , OUTPT064
     . A8,8H SOURCES,23X,1HI/2H I,94(1H-),1HI/                          OUTPT065
     . 37H I RECEPTOR I   RECEPTOR LOCATION    I,17X,24HEXPECTED ARITHMETOUTPT066
     .IC MEAN,18X,1HI/13H I  NUMBER  I,23X,1HI,59X,1HI/                 OUTPT067
     . 2H I,10(1H-),1HI,23(1H-),1HI,59(1H-),1HI/                        OUTPT068
     . 2H I,10X,1HI,5X,12H(KILOMETERS),6X,1HI,18X,22H(MICROGRAMS/CU. METOUTPT069
     .ER),19X,1HI,/2H I,10X,1HI,5X,1HX,5X,1HI,5X,1HY,                   OUTPT070
     . 5X,4(3HI  ,A8,1X),3HI  ,A8,2H I)                                 OUTPT071
 210  FORMAT(2H I,I6,4X,2(1HI,F9.3,2X),1HI,5(1PE10.3,2H I))             OUTPT072
 220  FORMAT(1H1,9X,20A4/10H  MONTH = ,A4,12H   PERIOD = ,A4,4H TO ,    OUTPT073
     . A4,16H HOURS ON A WEEK,A4)                                       OUTPT074
 260  FORMAT(2HI,10(1H-),1HI,7(11(1H-),1HI))                            OUTPT075
      RETURN                                                            OUTPT076
      END                                                               OUTPT077
```

SUBROUTINE PLRISE

Purpose:

To calculate the effective height and the vertical and horizontal dispersion coefficients for a given stack.

Input:

The stack parameters and current meteorological conditions.

Output:

1. The effective height, h_{eff}.
2. The vertical and horizontal dispersion coefficients, σ_{yo} and σ_{zo}.
3. KSTAB, a flag used in the TRAN function
   - = 0, the modified stack height is below the lid
   - = 1, the modified stack height is initially above the lid
   - = 2, the plume will penetrate the lid.

Procedure:

1. For point sources having no plume rise:

$$h_{eff} = \max(Z_S, H_B, \Delta Z/2.)$$

$$\sigma_{yo} = \Delta Y/2.4$$

$$\sigma_{zo} = \Delta Z/2.4$$

KSTAB = 0 or 1

2. For point sources which may undergo plume rise:

   a. Estimate the wind speed at the top of the aerovane

   b. Modify the stack height by the effect of the stack downwash

   c. Test for building downwash effects. If downwash occurs:

$$h_{eff} = H_B + .5L_B$$

$$\sigma_{yo} = \sigma_{zo} = h_{eff}/1.2$$

KSTAB = 0 or 1

   d. Test to determine if the buoyant plume rise is significant.

e. Check for an inversion

f. Compute the plume rise using function RISE

g. If no downwash occurs:

$$H_{eff} = Z_S + 2\left(\frac{VS}{U_a} - 1.5\right)\cdot DS + \text{plume rise}$$

$$\sigma_{yo} = \Delta Y/2.4$$

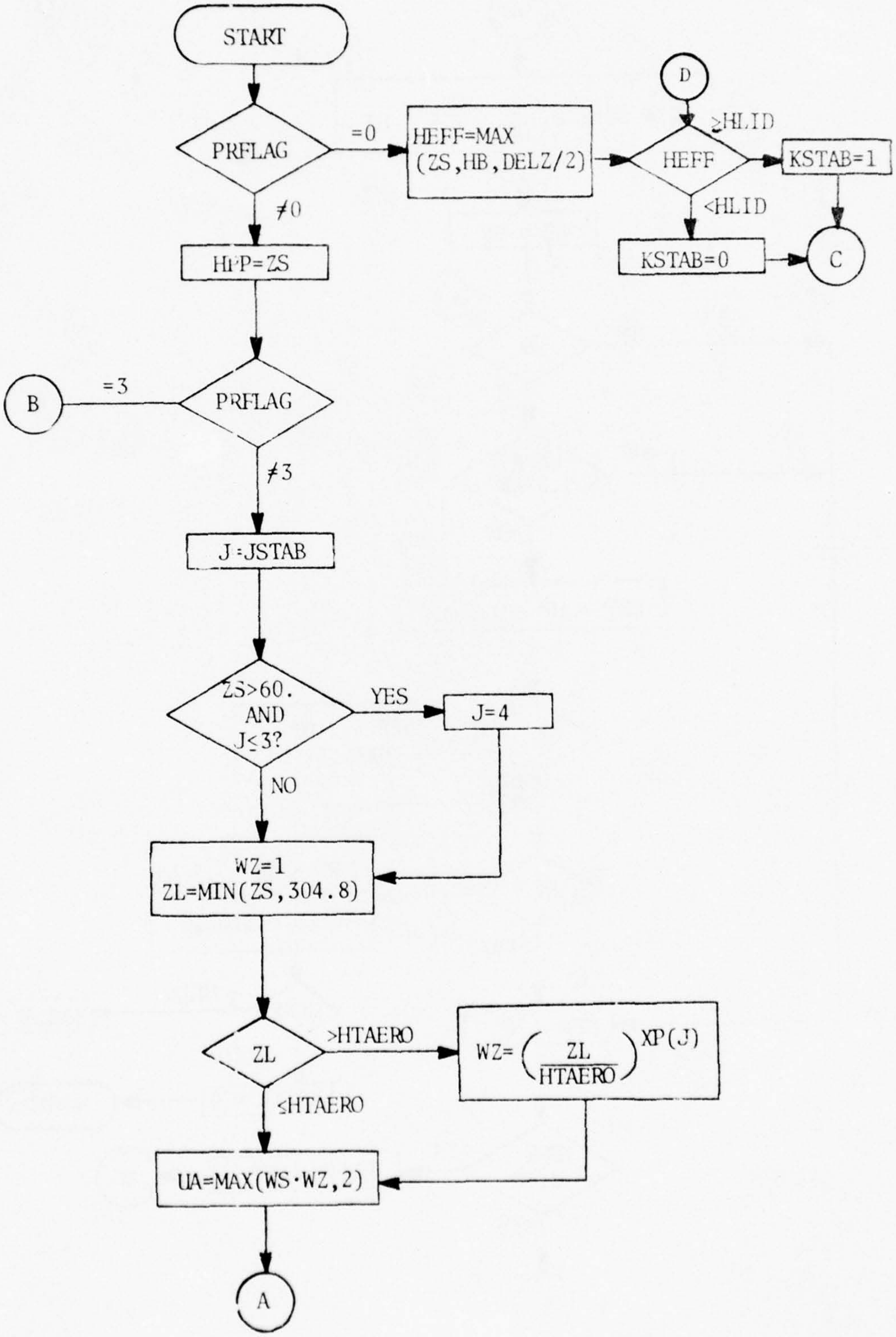
$$\sigma_{zo} = \Delta Z/2.4$$

KSTAB = 1 or 2

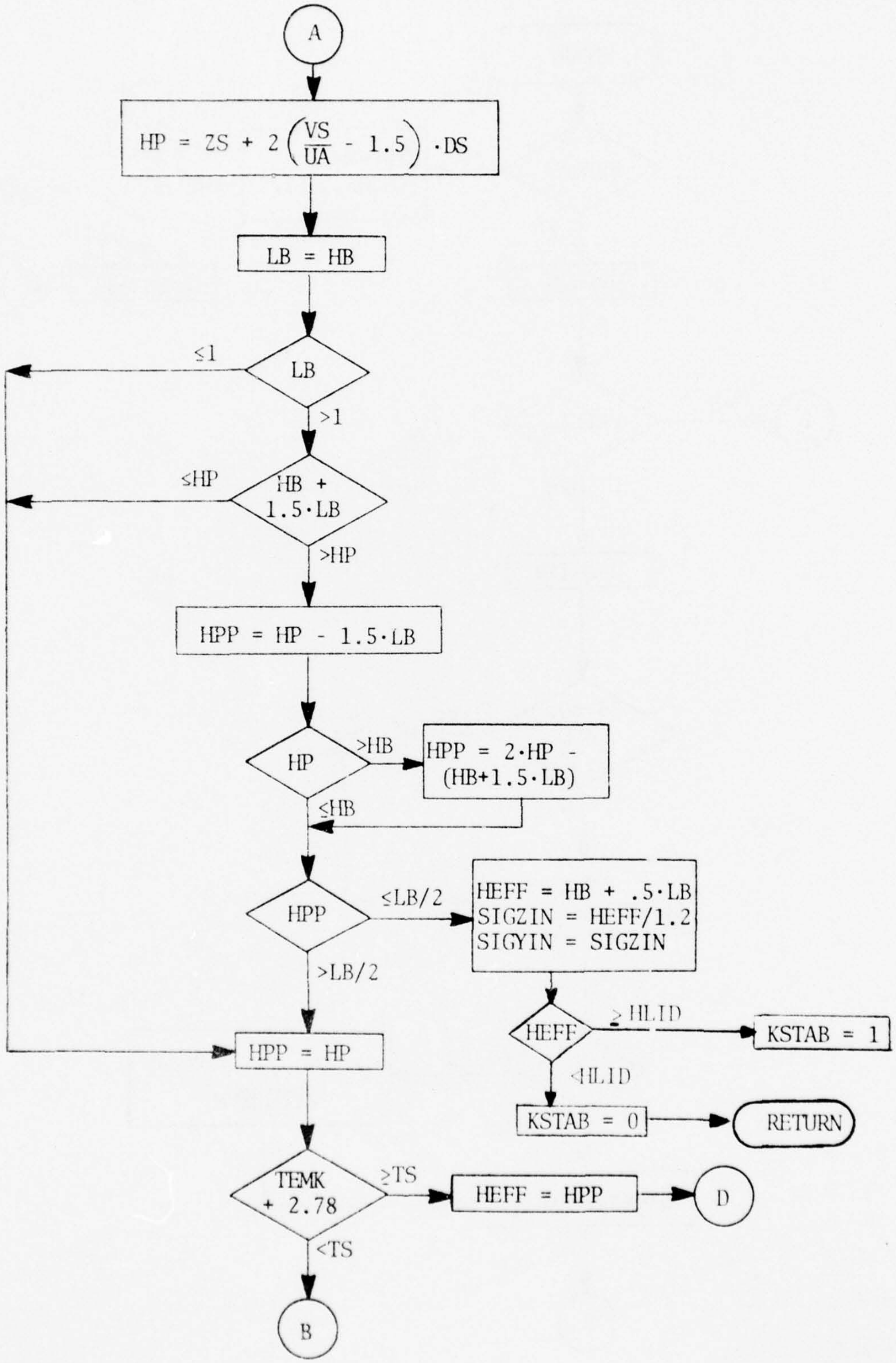
Functions <u>Called</u>:

RISE

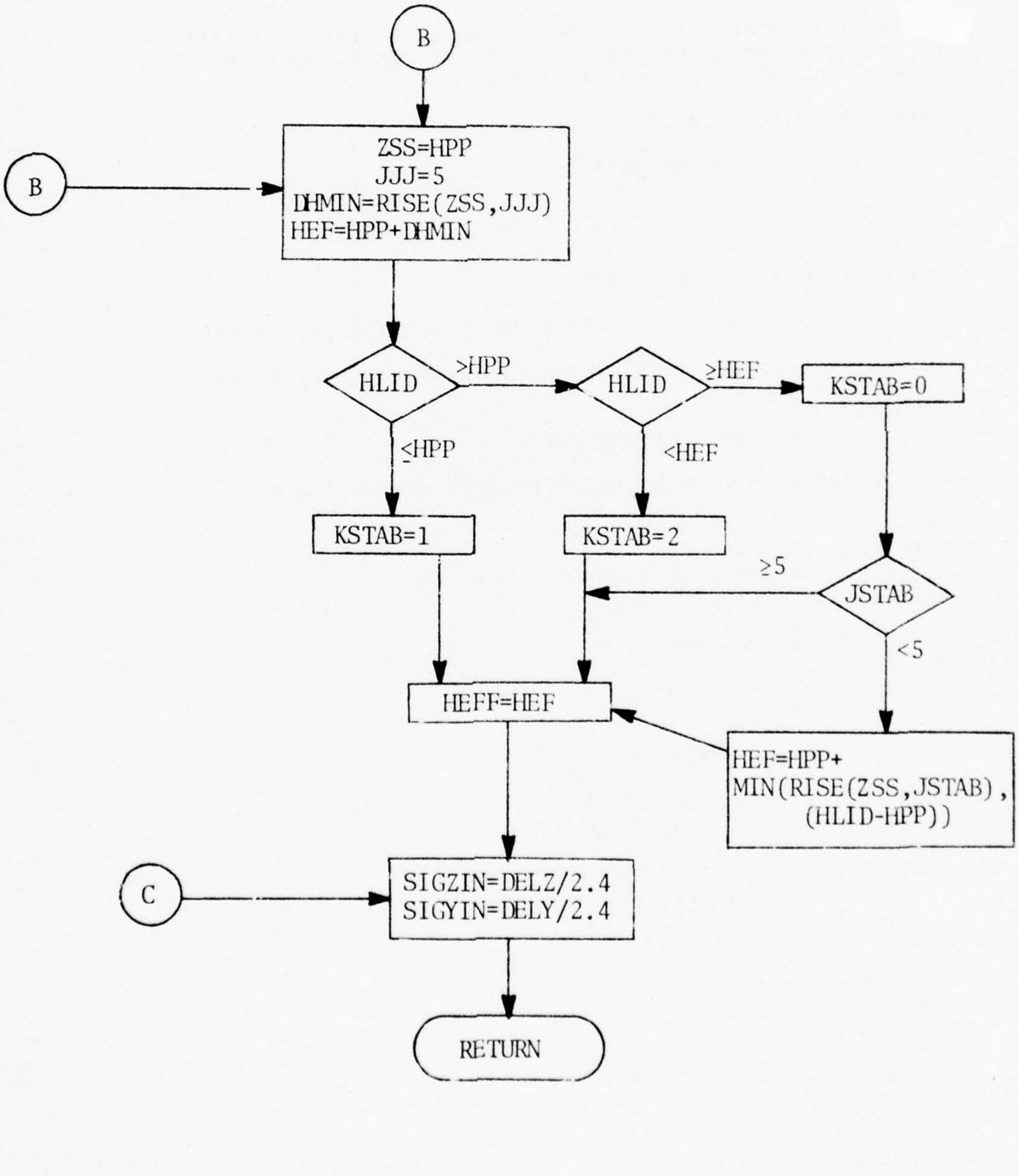
SUBROUTINE PLRISE



SUBROUTINE PLRISE (Cont'd.)



SUBROUTINE PLRISE (Cont'd.)



```
      SUBROUTINE PLRISE(HEFF,KSTAB,SIGZIN,SIGYIN)
C
C   THIS SUBROUTINE CALCULATES THE EFFECTIVE HEIGHT AND THE
C   VERTICAL AND HORIZONTAL DISPERSION COEFFICIENTS
C   FOR A GIVEN STACK
C
      REAL LB
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
     . TEMK,UA
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAFRO,XS,YS,ZS,DELY,DELZ,
     .  TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      COMMON /WNDPRO/ XP(6)
      IF (PRFLAG.NE.0) GO TO 100
C
C   FOR AN AREA SOURCE WITH A DIAMETER OF LESS THAN 50 METERS
C   THE EFFECTIVE EMISSION HEIGHT IS SET TO THE MAXIMUM OF
C   Z, THE BUILDING HEIGHT OR DELTA Z/2.0
C
      HEFF=AMAX1(ZS,HB,DELZ/2.)
   50 KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      GO TO 230
C
  100 CONTINUE
      HPP=ZS
      IF (PRFLAG.EQ.3) GO TO 130
C
C   FIRST TEST FOR DOWNWASH, THEN COMPUTE PLUME RISE, IF ANY
C
C   FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW
C
      J=JSTAB
      IF (ZS.GT.60..AND.J.LE.3) J=4
C
C   COMPUTE THE WINDSPEED AT THE ELEVATION OF THE STACK
C
      WZ=1.0
      ZL=AMIN1(ZS,304.8)
      IF (ZL.GT.HTAFRO) WZ=(ZL/HTAFRO)**XP(J)
      UA=AMAX1(WS*WZ,2.0)
C
C   COMPUTE STACK DOWNWASH
C
      HE=ZS+2.0*(VS/UA-1.5)*DS
      LB=HB
C
C   BUILDING DOWNWASH TESTS
C
      IF (LB.LE.1.) GO TO 110
      IF (HE.GE.(HB+1.5*LB)) GO TO 110
      HPP=HE-1.5*LB
      IF (HP.GT.HB) HPP=2.0*HP-(HB+1.5*LB)
      IF (HPP.GT.(LB/2.0)) GO TO 110
C
C   BUILDING DOWNWASH OCCURS
C
      HEFF=HB+0.5*LB
      SIGZIN=HEFF/1.2
      SIGYIN=SIGZIN
      KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      RETURN
```

```
C
C    NO BUILDING DOWNWASH, TEST FOR PLUME RISE
C
  110 HPP=HP
      IF (TS.GT.(TEMK+2.78)) GO TO 130
C
C    COLD PLUME
C
      HEFF=HPP
      GO TO 50
  130 CONTINUE
C
C    PLUME RISE EXPECTED TO BE SIGNIFICANT
C    CALCULATE MINIMUM PLUME RISE
C
      ZSS=HPP
      JJJ=5
      DHMIN=RISE(ZSS,JJJ)
      HEF=HPP+DHMIN
C
C    TEST FOR INTERFERENCE OF LID WITH MODIFIED PHYSICAL STACK
C    HEIGHT AND PLUME
C
      IF (HLID.GT.HPP) GO TO 220
C
C    LID INTERFERES WITH STACK HEIGHT, USE STABILITY 5 WITH
C    INFINITE LID HEIGHT
C
      KSTAB=1
      GO TO 225
C
C    LID INTERFERES WITH PLUME, USE STABILITY 5 WITH INFINITE LID
C
  220 IF (HLID.GE.HEF) GO TO 221
      KSTAB=2
      GO TO 225
C
C    CALCULATE PLUME RISE, PLUME CANNOT PENETRATE THE LID
C
  221 KSTAB=0
      IF (JSTAB.LT.5) HEF=HPP+AMIN1(RISE(ZSS,JSTAB),(HLID-HPP))
  225 CONTINUE
      HEFF=HEF
  230 SIGZIN=DELZ/2.4
      SIGYIN=DELY/2.4
      RETURN
      END
```

SUBROUTINE POLUT

Purpose:

To calculate the pollutant concentrations from point and area sources.

Input:

1. Source parameters for the current point or area.
2. Receptor locations.
3. Meteorological conditions.

Output:

Accumulated pollutant concentrations at all receptors for all stability classes.

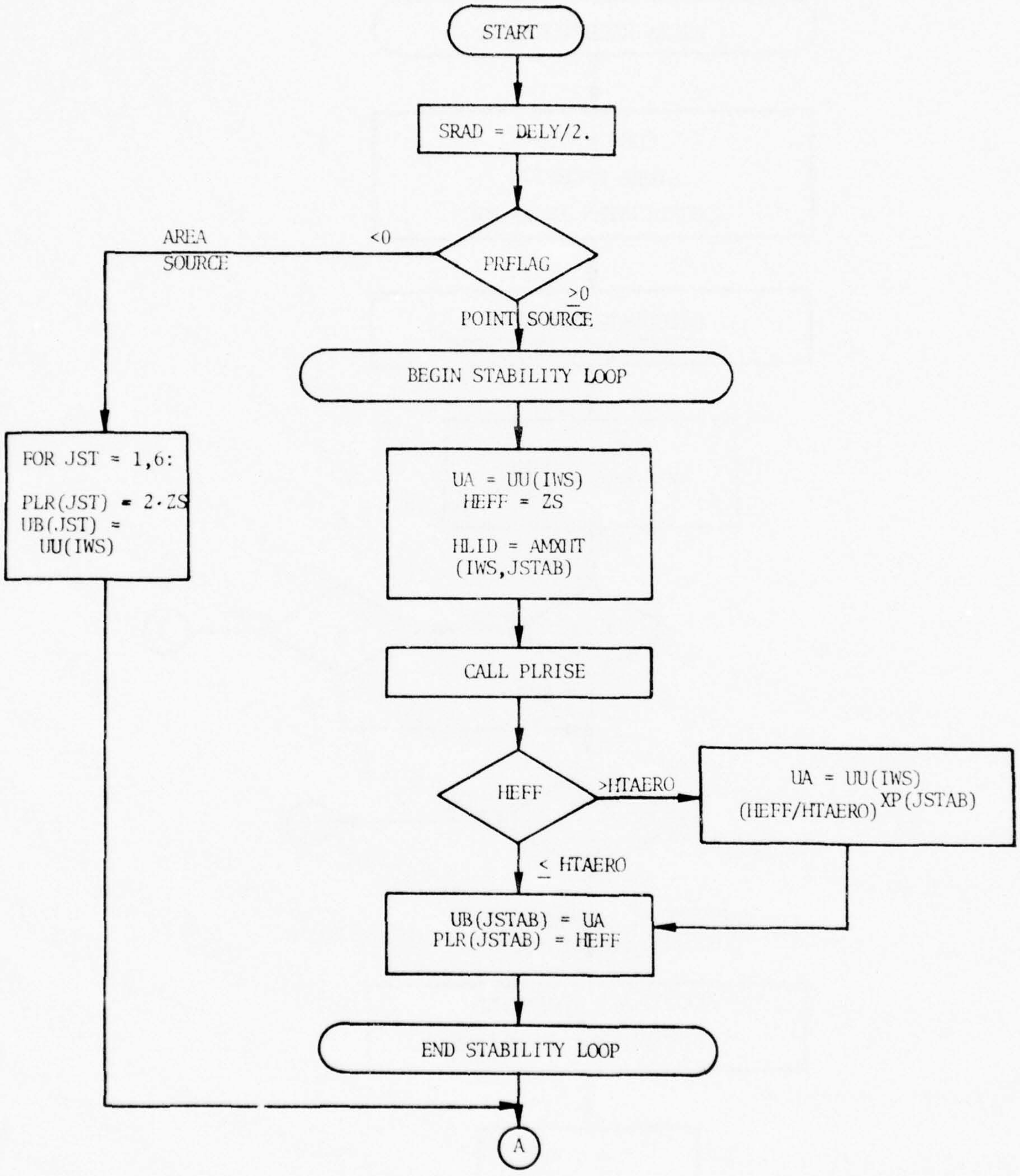
Procedure:

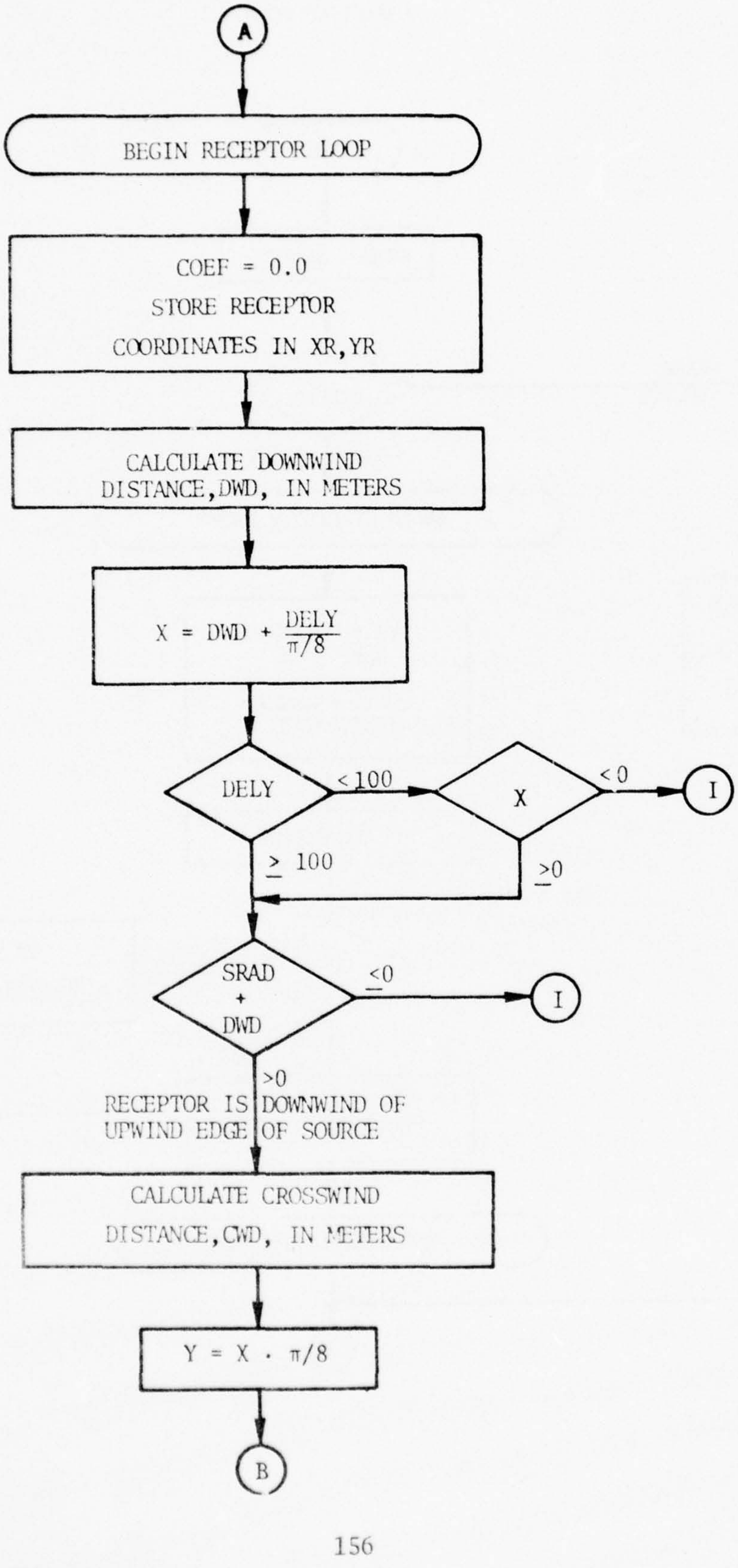
1. For all stability classes, determine the plume rise and wind speed at the height of the source.
2. Calculate the downwind and crosswind distances.
3. Calculate the distance from the virtual point source to the receptor by fitting an isosceles triangle of central angle 22.5^o to the source.
4. Check the position of the receptor relative to the source.
5. Determine if the receptor is inside, outside, or close to the source.
6. For all stability classes, find the critical distance and determine the effect of the lid height.
7. Accumulate the concentrations by receptor, pollutant and source type.

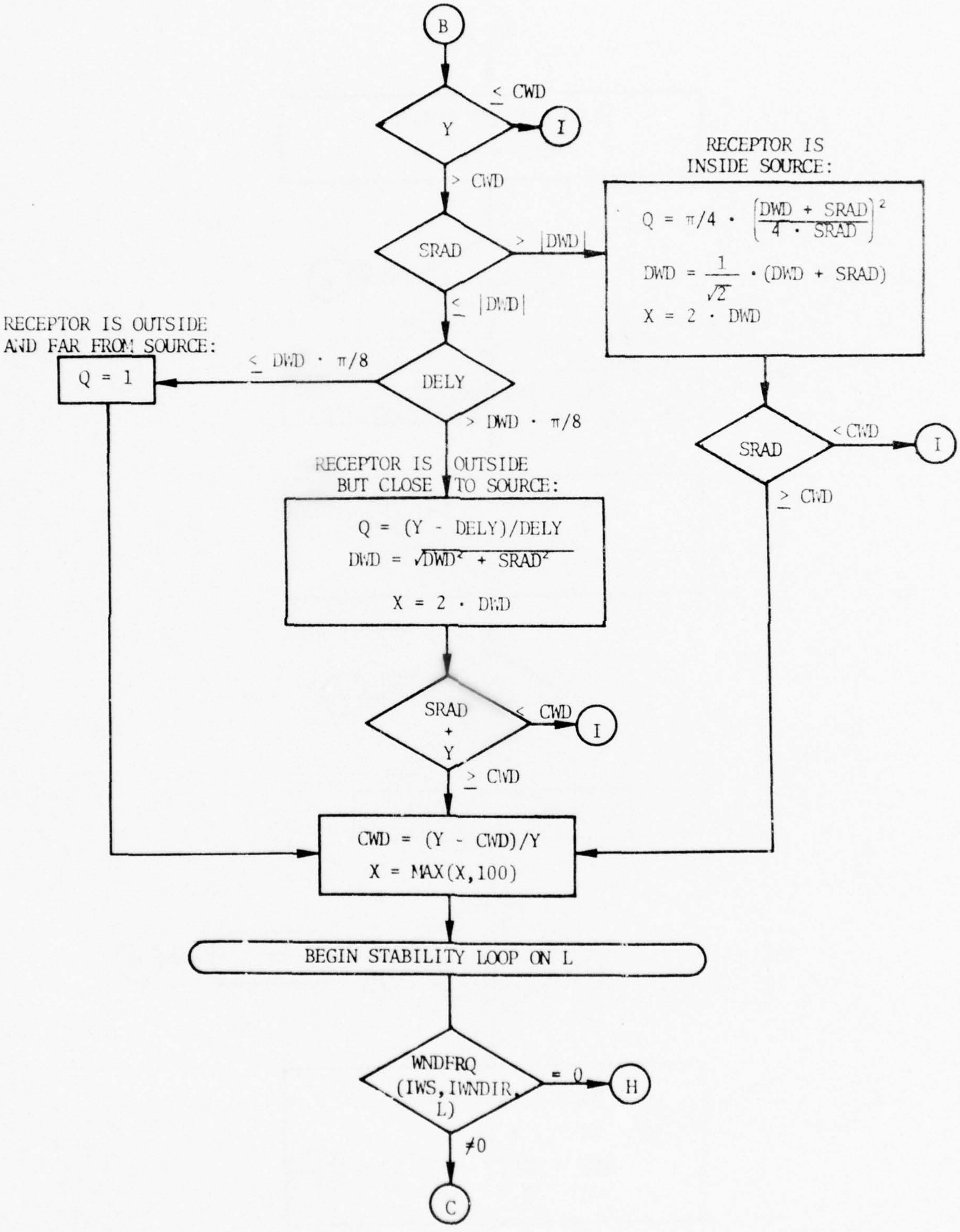
Subroutines Called:

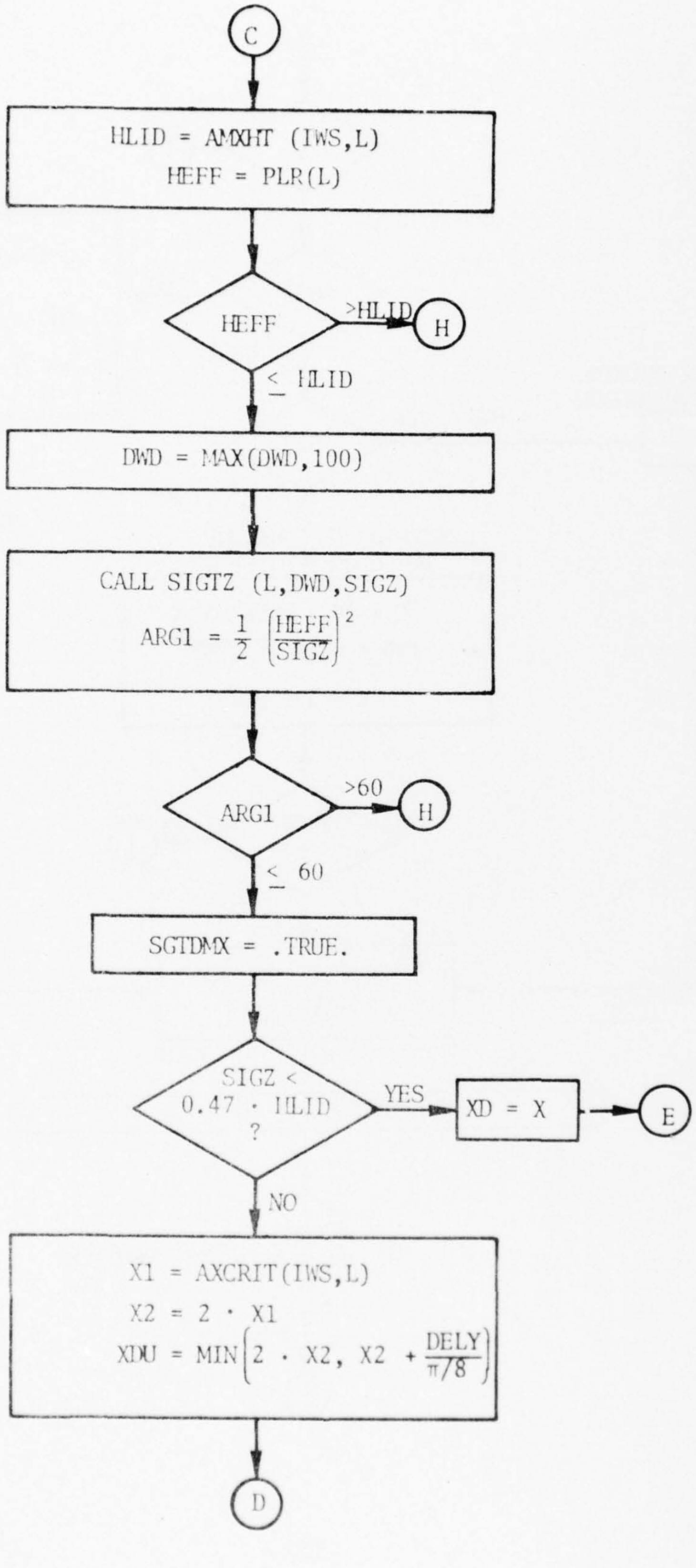
AREAWT, PLRISE, SIGTZ

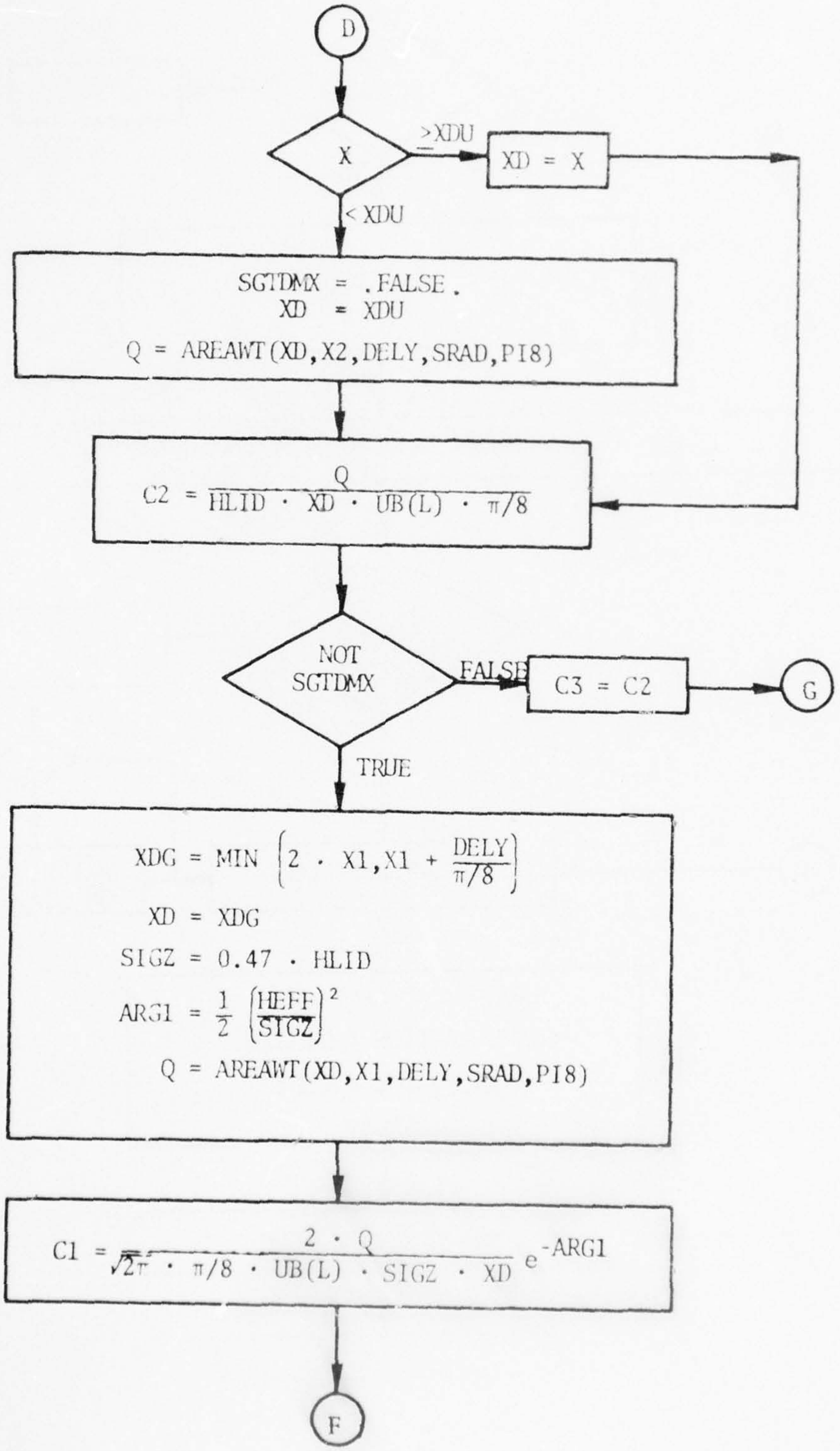
SUBROUTINE POLUT

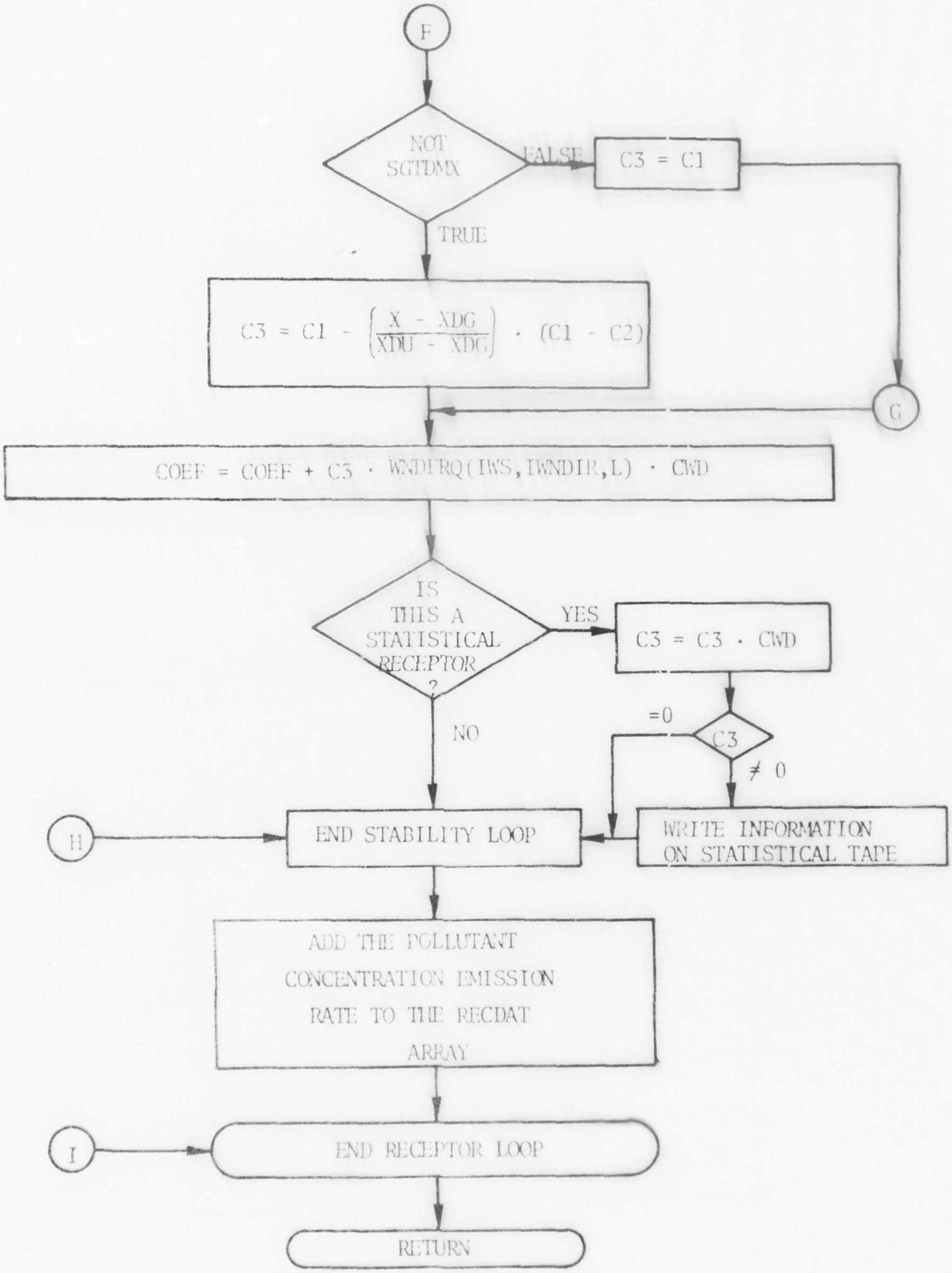



A
BEGIN RECEPTOR LOOP
COEF = 0.0
STORE RECEPTOR
COORDINATES IN XR,YR
CALCULATE DOWNWIND
DISTANCE,DWD, IN METERS
X = DWD + DELY/π/8
DELY
< 100
X
< 0
I
≥ 100
≥0
SRAD
+
DWD
≤0
I
>0
RECEPTOR IS DOWNWIND OF
UPWIND EDGE OF SOURCE
CALCULATE CROSSWIND
DISTANCE,CWD, IN METERS
Y = X · π/8
B


B
Y
≤ CWD
I
> CWD
SRAD
> |DWD|
≤ |DWD|
RECEPTOR IS INSIDE SOURCE:
Q = π/4 · ((DWD + SRAD)/(4 · SRAD))²
DWD = 1/√2 · (DWD + SRAD)
X = 2 · DWD
SRAD
< CWD
I
≥ CWD
RECEPTOR IS OUTSIDE AND FAR FROM SOURCE:
Q = 1
≤ DWD · π/8
DELY
> DWD · π/8
RECEPTOR IS OUTSIDE BUT CLOSE TO SOURCE:
Q = (Y - DELY)/DELY
DWD = √DWD² + SRAD²
X = 2 · DWD
SRAD + Y
< CWD
I
≥ CWD
CWD = (Y - CWD)/Y
X = MAX(X,100)
BEGIN STABILITY LOOP ON L
WNDFRQ (IWS,IWNDIR, L)
= 0
H
≠0
C


C
HLID = AMXHT (IWS,L)
HEFF = PLR(L)
HEFF
>HLID
H
≤ HLID
DWD = MAX(DWD,100)
CALL SIGTZ (L,DWD,SIGZ)
$ARG1 = \frac{1}{2}\left(\frac{HEFF}{SIGZ}\right)^2$
ARG1
>60
H
≤ 60
SGTDMX = .TRUE.
SIGZ < 0.47 · HLID ?
YES
XD = X
E
NO
X1 = AXCRIT(IWS,L)
X2 = 2 · X1
$XDU = MIN\left(2 \cdot X2,\ X2 + \frac{DELY}{\pi/8}\right)$
D


D
X
>XDU
XD = X
< XDU
SGTDMX = .FALSE.
XD = XDU
Q = AREAWT(XD,X2,DELY,SRAD,PI8)
$C2 = \frac{Q}{HLID \cdot XD \cdot UB(L) \cdot \pi/8}$
NOT SGTDMX
FALSE
C3 = C2
G
TRUE
$XDG = MIN \left\{2 \cdot X1, X1 + \frac{DELY}{\pi/8}\right\}$
XD = XDG
SIGZ = 0.47 · HLID
$ARG1 = \frac{1}{2} \left(\frac{HEFF}{SIGZ}\right)^2$
Q = AREAWT(XD,X1,DELY,SRAD,PI8)
$C1 = \frac{2 \cdot Q}{\sqrt{2\pi} \cdot \pi/8 \cdot UB(L) \cdot SIGZ \cdot XD} e^{-ARG1}$
F


F
NOT SGTDMX
FALSE
C3 = C1
TRUE
C3 = C1 - (X - XDG / XDU - XDG) · (C1 - C2)
G
COEF = COEF + C3 · WNDFRQ(IWS,IWNDIR,L) · CWD
IS THIS A STATISTICAL RECEPTOR ?
YES
C3 = C3 · CWD
=0
C3
≠ 0
NO
H
END STABILITY LOOP
WRITE INFORMATION ON STATISTICAL TAPE
ADD THE POLLUTANT CONCENTRATION EMISSION RATE TO THE RECDAT ARRAY
I
END RECEPTOR LOOP
RETURN

```
      SUBROUTINE POLUT                                                  POLUT000
C                                                                       POLUT001
C     THIS ROUTINE CALCULATES POLLUTANT CONCENTRATIONS FROM POINT       POLUT002
C     AND AREA SOURCES                                                  POLUT003
C                                                                       POLUT004
      LOGICAL SGTDMX                                                    POLUT005
      COMMON /AIRQAL/ RECDAT(3, 6,312)                                  POLUT006
      COMMON /ANNMET/ TBAR,ADD,PA,PAX,WSBAR,DTBAR                       POLUT007
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)             POLUT008
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,XS,YS,ZS,DELY,DELZ,      POLUT009
     . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL                                  POLUT010
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,   POLUT011
     . TEMK,UA                                                          POLUT012
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)          POLUT013
      COMMON /MONMET/ TMBAR,WSMBAR,DPTHMX,DMX                           POLUT014
      COMMON /RCPT/ NRECEP,RECEP(2,312)                                 POLUT015
      COMMON /WNDPRO/ XP(6)                                             POLUT016
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)               POLUT017
      DIMENSION PLR(6),UB(6)                                            POLUT018
C                                                                       POLUT019
      SRAD=DELY/2.                                                      POLUT020
C                                                                       POLUT021
C     PRFLAG LESS THAN 0.0 INDICATES AN AREA SOURCE                     POLUT022
C                                                                       POLUT023
      IF (PRFLAG.GE.0.0) GO TO 2                                        POLUT024
      DO 1 JST=1,6                                                      POLUT025
      PLR(JST)=2.*ZS                                                    POLUT026
    1 UB(JST)=UU(IWS)                                                   POLUT027
      GO TO 4                                                           POLUT028
    2 DO 3 JSTAB=1,6                                                    POLUT029
      UA=UU(IWS)                                                        POLUT030
      HEFF=ZS                                                           POLUT031
      HLID=AMXHT(IWS,JSTAB)                                             POLUT032
      CALL PLRISE(HEFF,KSTAB,SIGZIN,SIGYIN)                             POLUT033
      IF (HEFF.GT.HTAERO) UA=UU(IWS)*(HEFF/HTAERO)**XP(JSTAB)           POLUT034
      UB(JSTAB)=UA                                                      POLUT035
    3 PLR(JSTAB)=HEFF                                                   POLUT036
    4 CONTINUE                                                          POLUT037
C                                                                       POLUT038
C     BEGIN RECEPTOR LOOP                                               POLUT039
C                                                                       POLUT040
      DO 700 IRECEP=1,NRECEP                                            POLUT041
      COEF=0.0                                                          POLUT042
      XR=RECEP(1,IRECEP)                                                POLUT043
      YR=RECEP(2,IRECEP)                                                POLUT044
C                                                                       POLUT045
C     COMPUTE DOWNWIND AND CROSSWIND DISTANCES AND CHECK                POLUT046
C     POSITION OF RECEPTOR RELATIVE TO SOURCE                           POLUT047
C                                                                       POLUT048
      DWD=1000.*((YS-YR)*COSEWD+(XS-XR)*SINEWD)                         POLUT049
      X=DWD+DELY/PI8                                                    POLUT050
      IF (DELY.GE.100.) GO TO 110                                       POLUT051
      IF (X.LT.0.0) GO TO 700                                           POLUT052
  110 IF (SRAD+DWD.LE.0.) GO TO 700                                     POLUT053
  120 CWD=1000.*ABS((YS-YR)*SINEWD-(XS-XR)*COSEWD)                      POLUT054
      Y=X*PI8                                                           POLUT055
      IF (Y.LE.CWD) GO TO 700                                           POLUT056
      IF (SRAD.GT.ABS(DWD)) GO TO 130                                   POLUT057
      IF (DELY.LE.DWD*PI8) GO TO 140                                    POLUT058
C                                                                       POLUT059
C     RECEPTOR IS OUTSIDE BUT CLOSE TO SOURCE                           POLUT060
C                                                                       POLUT061
```

```
      Q=(Y-DELY)/DELY
      DWD=SQRT(DWD**2+SRAD**2)
      X=2.*DWD
      IF (CWD-(SRAD+Y)) 150,150,700
C
C   RECEPTOR IS INSIDE SOURCE
C
  130 Q=PI4*((DWD+SRAD)/(4.*SRAD))**2
      DWD=.70711*(DWD+SRAD)
      X=DWD*2.
      IF (CWD-SRAD) 150,700,700
C
C   RECEPTOR IS OUTSIDE AND FAR FROM SOURCE
C
  140 Q=1.
C
C   COMPUTE LINEAR WEIGHTING FACTOR
C
  150 CWD=(Y-CWD)/Y
      IF(X.LT.100.)X=100.
C
C   BEGIN STABILITY CLASS LOOP
C
      DO 550 L=1,6
      IF(WNDFRQ(IWS,IWNDIR,L).EQ.0.0)GO TO 550
      HLID=AMXHT(IWS,L)
      HEFF=PLR(L)
      IF (HEFF.GT.HLID) GO TO 550
      IF (DWD.LT.100.) DWD=100.
C
C   COMPUTE THE VERTICAL DISPERSION COEFFICIENT
C
 5526 CALL SIGTZ(L,DWD,SIGZ)
      ARG1=.5*(HEFF/SIGZ)**2
      IF (ARG1.GT.60.) GO TO 550
      SGTDMX=.TRUE.
      IF (SIGZ.LT.0.47*HLID) GO TO 5527
 5528 X1=AXCRIT(IWS,L)
      X2=2.*X1
      XDU=AMIN1(2.*X2,X2+DELY/PI8)
      IF(X.GE.XDU) GO TO 5531
      SGTDMX=.FALSE.
      XD=XDU
      Q=AREAWT(XD,X2,DELY,SRAD,PI8)
      GO TO 5532
C
C   UNIFORM MIXING IS ASSUMED
C
 5531 XD=X
 5532 C2=Q/(HLID*XD*UB(L)*PI8)
      IF (.NOT.SGTDMX) GO TO 5533
      C3=C2
      GO TO 5000
 5533 XDG=AMIN1(2.*X1,X1+DELY/PI8)
      XD=XDG
      SIGZ=0.47*HLID
      ARG1=.5*(HEFF/SIGZ)**2
      Q=AREAWT(XD,X1,DELY,SRAD,PI8)
      GO TO 5534
 5527 XD=X
 5534 C1=Q*2.03179635/(UB(L)*SIGZ*XD)*EXP(-ARG1)
      IF(.NOT.SGTDMX) GO TO 5555
      C3=C1
      GO TO 5000
C
C   INTERPOLATE BETWEEN C1 AT XDG AND C2 AT XDU
C
 5555 CONTINUE
      C3=C1-((X-XDG)/(XDU-XDG))*(C1-C2)
 5000 CONTINUE
C
C   APPLY WNDFRQ AND LINEAR WEIGHTING FACTORS
C
      COEF=COEF+C3*WNDFRQ(IWS,IWNDIR,L)*CWD
C
      IF (IRSTAT(IRECEP).EQ.0) GO TO 550
C
C   IF STATISTICAL OPTION IS CHOSEN RECORD COUPLING
C   COEFFICIENT DUE TO THIS WIND DIRECTION, SPEED AND
C   STABILITY CLASS ON NSTAPE
C
      C3=C3*CWD
      IF (C3.EQ.0.0) GO TO 550
      WRITE (NSTAPE) IWNDIR,WD,IWS,WS,ITYPE,(EMIS(K),K=1,NPOL),
     . IRSTAT(IRECEP),L,C3,HLID
  550 CONTINUE
  600 CONTINUE
C
C   ADD EMISSIONS TIMES COUPLING COEFFICIENT TO CONCENTRATIONS
C
      DO 800 IPOL=1,NPOL
      RECDAT(ITYPE,IPOL,IRECEP)=RECDAT(ITYPE,IPOL,IRECEP)
     . +COEF*EMIS(IPOL)
  800 CONTINUE
  700 CONTINUE
      RETURN
      END
```

SUBROUTINE POLUTL

Purpose:

To prepare the data required by the line source model and, for each receptor, to call the model for each stability class and then add the pollutant concentrations calculated to the accumulated totals at that receptor.

Input:

1. Source parameters for the current line.
2. Wind speed and direction.
3. Wind frequency, critical distance and mixing height arrays.
4. Virtual horizontal and vertical distances from the source to the pseudo upwind point source for all stability classes.

Output:

Accumulated pollutant concentrations at all receptors for all stability classes.

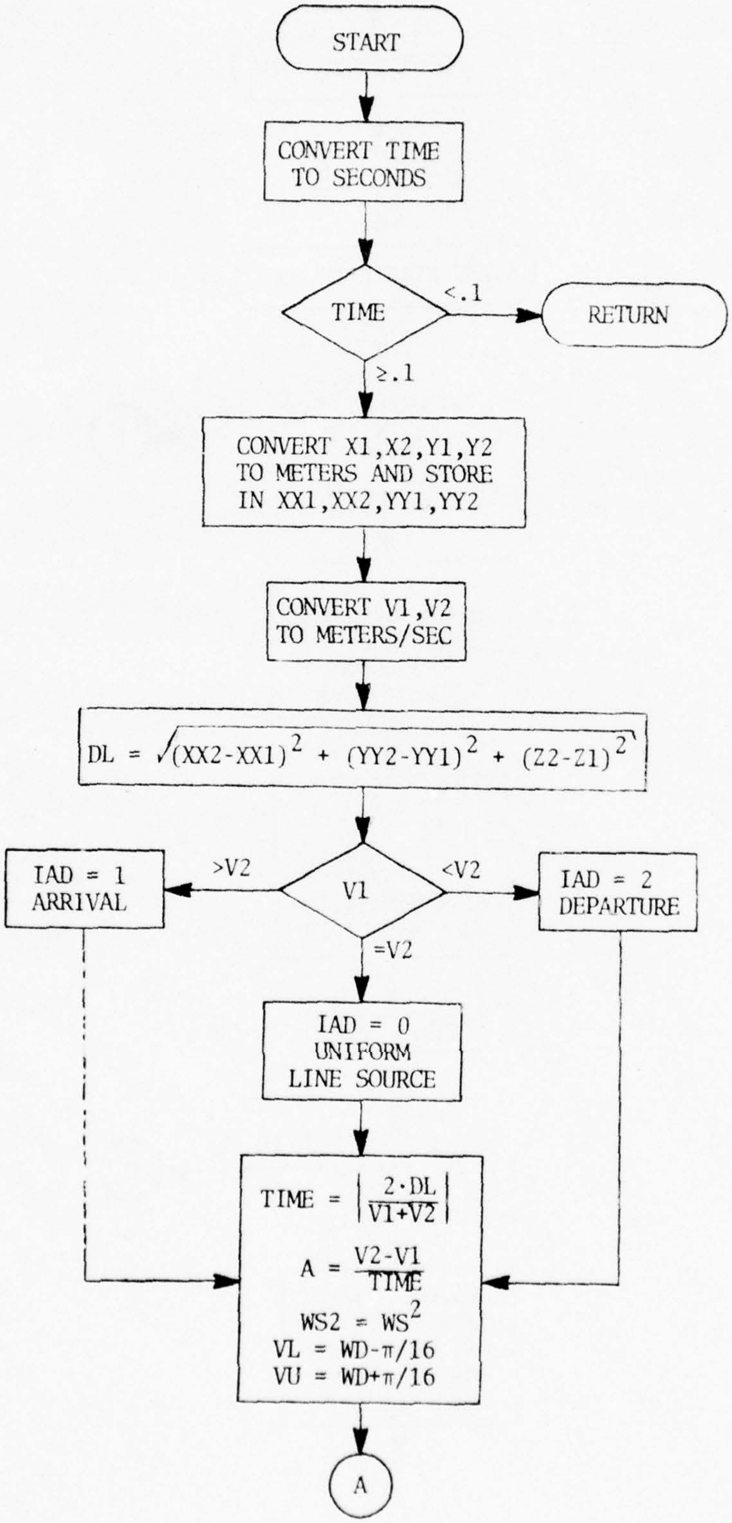
Procedure:

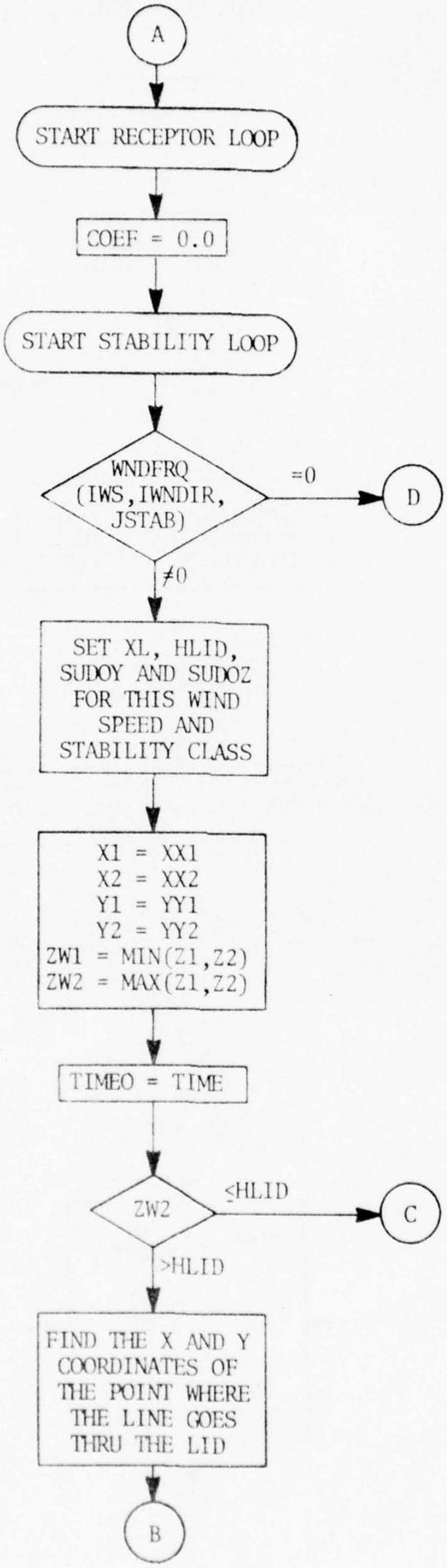
1. Convert source data to proper units.
2. Calculate variables used by the line source model.
3. For each receptor, loop through all stability classes and use the three or four point Gaussian quadrature procedure to call the line model for uniform or non-uniform lines.
4. Accumulate the concentrations by receptor, pollutant and source type.

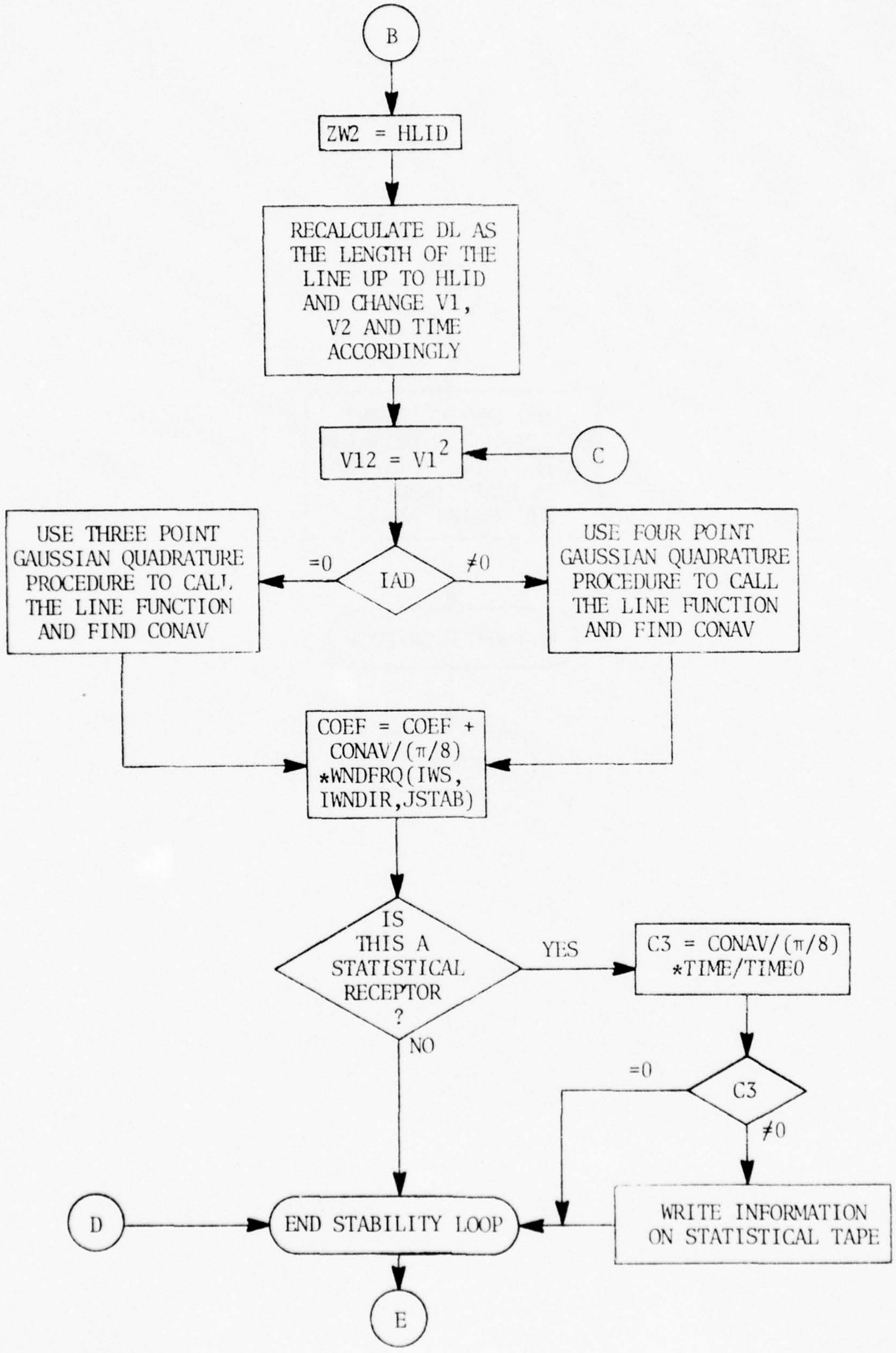
Subroutines Called:

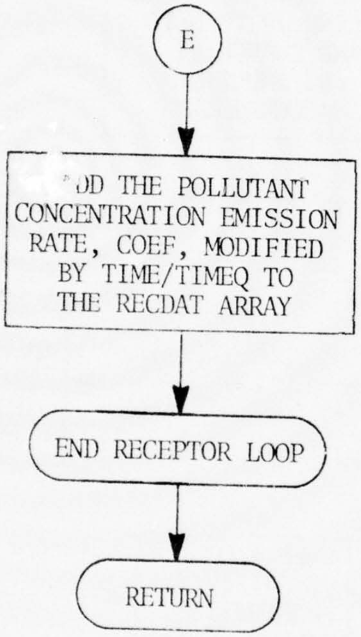
QG3, QG4, AINE

SUBROUTINE POLUTL


START
CONVERT TIME TO SECONDS
TIME
<.1
RETURN
≥.1
CONVERT X1,X2,Y1,Y2 TO METERS AND STORE IN XX1,XX2,YY1,YY2
CONVERT V1,V2 TO METERS/SEC
DL = $\sqrt{(XX2-XX1)^2 + (YY2-YY1)^2 + (Z2-Z1)^2}$
V1
>V2
IAD = 1 ARRIVAL
<V2
IAD = 2 DEPARTURE
=V2
IAD = 0 UNIFORM LINE SOURCE
TIME = $\left|\frac{2 \cdot DL}{V1+V2}\right|$
A = $\frac{V2-V1}{TIME}$
WS2 = WS^2
VL = WD-π/16
VU = WD+π/16
A


A
START RECEPTOR LOOP
COEF = 0.0
START STABILITY LOOP
WNDFRQ (IWS,IWNDIR, JSTAB)
=0
D
≠0
SET XL, HLID, SUDOY AND SUDOZ FOR THIS WIND SPEED AND STABILITY CLASS
X1 = XX1
X2 = XX2
Y1 = YY1
Y2 = YY2
ZW1 = MIN(Z1,Z2)
ZW2 = MAX(Z1,Z2)
TIMEO = TIME
ZW2
≤HLID
C
>HLID
FIND THE X AND Y COORDINATES OF THE POINT WHERE THE LINE GOES THRU THE LID
B


B
ZW2 = HLID
RECALCULATE DL AS THE LENGTH OF THE LINE UP TO HLID AND CHANGE V1, V2 AND TIME ACCORDINGLY
V12 = V1^2
C
USE THREE POINT GAUSSIAN QUADRATURE PROCEDURE TO CALL THE LINE FUNCTION AND FIND CONAV
=0
IAD
≠0
USE FOUR POINT GAUSSIAN QUADRATURE PROCEDURE TO CALL THE LINE FUNCTION AND FIND CONAV
COEF = COEF + CONAV/(π/8) *WNDFRQ(IWS, IWNDIR,JSTAB)
IS THIS A STATISTICAL RECEPTOR ?
YES
C3 = CONAV/(π/8) *TIME/TIME0
NO
=0
C3
≠0
D
END STABILITY LOOP
WRITE INFORMATION ON STATISTICAL TAPE
E


E
ADD THE POLLUTANT CONCENTRATION EMISSION RATE, COEF, MODIFIED BY TIME/TIMEQ TO THE RECDAT ARRAY
END RECEPTOR LOOP
RETURN

```
      SUBROUTINE POLUTL
C
C   THIS ROUTINE PREPARES DATA REQUIRED BY THE LINE SOURCE
C   MODEL AND CALLS THE MODEL TO DETERMINE CONCENTRATIONS AT ALL
C   RECEPTORS
C
      EXTERNAL AINE
      COMMON /AIRQAL/ RECDAT(3, 6,312)
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP,AA1,AA2,AA3,AA4,AA5,AA6
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
     . TEMK
      COMMON /METSET/ WNDFRQ(6,16,6),UU(6),SINWD(16),COSWD(16)
      COMMON /RCPT/ NRECEP,RECEP(2,312)
      COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)
      COMMON /XTRAN/ XL
      COMMON /WDUN/ WSAVE,SUDY(6),SUDZ(6)
C
C   CONVERT SOURCE DATA TO PROPER UNITS
C
      TIME = TIME * 3600.
      IF (TIME .LT. 0.1) GO TO 500
      XX1= X1 * 1000.
      XX2= X2 * 1000.
      YY1= Y1 * 1000.
      YY2= Y2 * 1000.
      V1 = V1 / 3.6
      V2 = V2 / 3.6
C
C   CALCUIATE VARIABLES USED BY THE LINE SOURCE MODEL
C
      DL=SQRT((XX2-XX1)**2+(YY2-YY1)**2+(Z2-Z1)**2)
      NPT=1
      IAD=0
      IF(V1.LT.(V2-.01))IAD =2
      IF(V1.GT.(V2+.01))IAD =1
      IF (IAD.NE.0) NPT=2
      TIME = ABS(2.*DL/(V1+V2))
      A=(V2-V1)/TIME
      WS2=WS*WS
      VL=WD-PI16
      VU=WD+PI16
C
C   BEGIN RECEPTOR LOOP
C
      DO 100 IRECEP=1,NRECEP
      COEF=0.0
C
C   BEGIN STABILITY LOOP
C
      DO 200 JSTAB=1,6
      IF(WNDFRQ(IWS,IWNDIR,JSTAB).EQ.0)GO TO 200
      XL=AXCRIT(IWS,JSTAB)
      HLID=AMXHT(IWS,JSTAB)
      SUDOY=SUDY(JSTAB)
      SUDOZ=SUDZ(JSTAB)
      X1=XX1
      X2=XX2
      Y1=YY1
      Y2=YY2
      ZW1=Z1
      ZW2=Z2
      IF(Z1.LE.Z2) GO TO 15
      ZW1=Z2
      ZW2=Z1
   15 TIMEO=TIME
      IF(ZW2.LE.HLID) GO TO 18
C
C   FIND POINT WHERE LINE GOES THRU THE LID AND CHANGE COORDINATES
C
      F=(HLID-ZW1)/(ZW2-ZW1)
      IF (Z1 .GT. Z2) GO TO 16
      X2 = X1+(X2-X1)*F
      Y2 = Y1+(Y2-Y1)*F
      GO TO 17
   16 X1 = X2+(X1-X2)*F
      Y1 = Y2+(Y1-Y2)*F
   17 ZW2 = HLID
C
C   RECALCULATE THE LENGTH OF THE LINE UP TO HLID AND
C   CHANGE VELOCITIES ACCORDINGLY
C
      DLSQ = (X1-X2)**2 + (Y1-Y2)**2 + (ZW1-ZW2)**2
      DL = SQRT(DLSQ)
      IF (Z2 .GT. Z1) V2 = SQRT(V1*V1+2.*A*DL)
      IF (Z2 .LT. Z1) V1 = SQRT(V2*V2-2.*A*DL)
      TIME = 2.* DL / (V1+V2)
   18 V12 = V1 * V1
C
C   CALL THE LINE FUNCTION TO DETERMINE POLLUTANT CONCENTRATION
C
      GO TO (203,204)     ,NPT
  203 CALL QG3(VL,VU,AINE,CONAV)
      GO TO 220
  204 CALL QG4(VL,VU,AINE,CONAV)
  220 CONTINUE
      COEF=COEF+CONAV/PI8 *WNDFRQ(IWS,IWNDIR,JSTAB)
      IF (IRSTAT(IRECEP).EQ.0) GO TO 200
C
C   IF STATISTICAL OPTION IS CHOSEN RECORD THE COUPLING
C   COEFFICIENT FOR THIS WIND DIRECTION, SPEED AND STABILITY CLASS
C   ON NSTAPE
C
      C3=CONAV/PI8 *TIME/TIMEO
      IF (C3.EQ.0.0) GO TO 200
      WRITE (NSTAPE) IWNDIR,WD,IWS,WS,ITYPE,(EMIS(K),K=1,NPOL),
     . IRSTAT(IRECEP),JSTAB,C3,HLID
  200 CONTINUE
C
C   ADD EMISSIONS TIMES COUPLING COEFFICIENT TO CONCENTRATIONS
C
      DO 300 IPOL=1,NPOL
  300 RECDAT(ITYPE,IPOL,IRECEP)=RECDAT(ITYPE,IPOL,IRECEP)
     . +EMIS(IPOL)*COEF * TIME / TIMEO
  100 CONTINUE
  500 CONTINUE
      RETURN
      END
```

SUBROUTINE PSEUDL

Purpose:

To call the SIGCY and SIGCZ functions to find the virtual distance in meters from the source to the pseudo upwind point source for all stability classes.

Input:

1. Wind speed
2. Initial dispersions in y and z directions

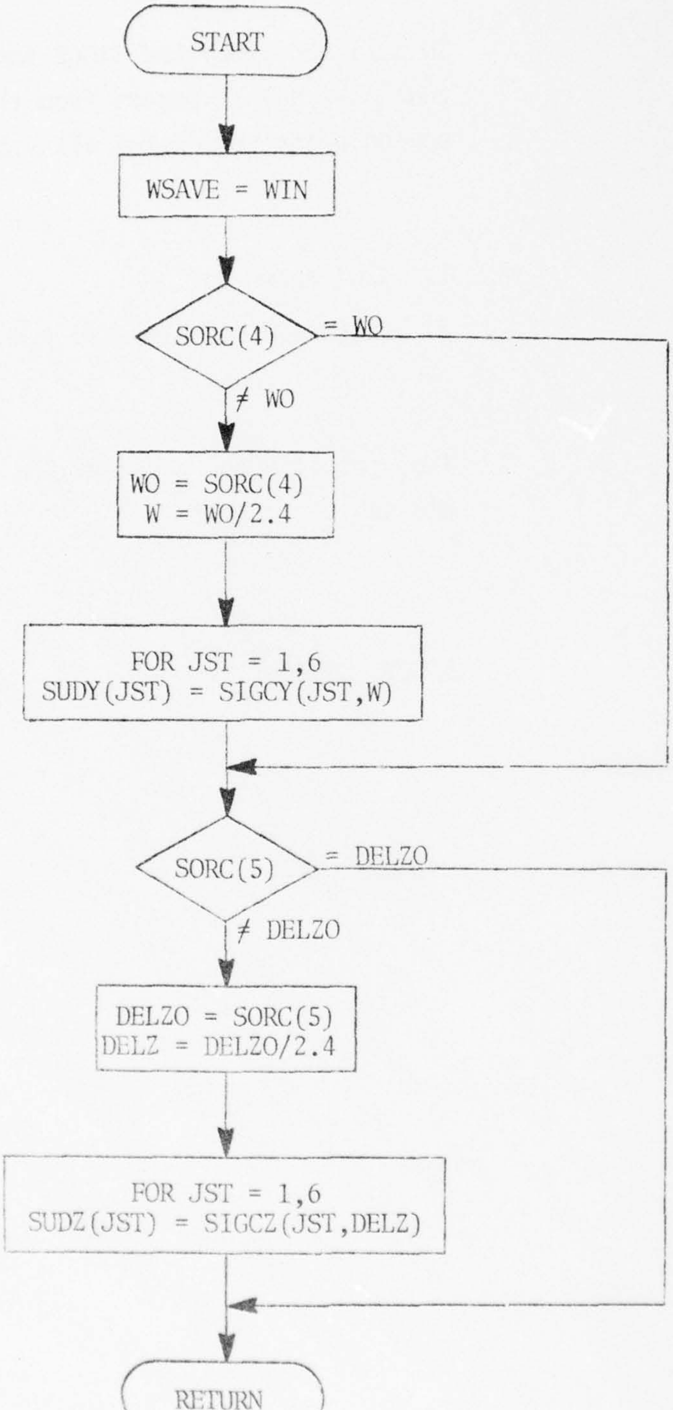
Output:

The virtual y and z directions for all stability classes.

Functions Called:

SIGCY, SIGCZ

SUBROUTINE PSEUDL(WIN)



```
      SUBROUTINE PSEUDL(WIN)                                            PSUDL000
C                                                                       PSUDL001
C   THIS ROUTINE CALL THE SIGCY AND SIGCZ FUNCTIONS TO FIND             PSUDL002
C   THE VIRTUAL DISTANCE FROM THE SOURCE TO THE PSEUDO UPWIND           PSUDL003
C   POINT SOURCE FOR ALL STABILITY CLASSES                              PSUDL004
C                                                                       PSUDL005
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,SORC(18),IPOL            PSUDL006
      COMMON /WDUN/ WSAVE,SUDY(6),SUDZ(6)                               PSUDL007
      DATA DELZO,WO /-1.,-1./                                           PSUDL008
C                                                                       PSUDL009
C   SAVE THE INPUT WIND SPEED                                           PSUDL010
C                                                                       PSUDL011
      K=0                                                               PSUDL012
      IF (WSAVE.NE.WIN) K=1                                             PSUDL013
      WSAVE=WIN                                                         PSUDL014
C                                                                       PSUDL015
C   IF SORC(4) OR SORC(5) ARE DIFFERENT FROM PREVIOUS VALUES,           PSUDL016
C   CALL THE SIG FUNCTIONS TO FIND DISTANCES IN METERS                  PSUDL017
C                                                                       PSUDL018
      IF (SORC(4).EQ.WO.AND.K.EQ.0) GO TO 20                            PSUDL019
      WO=SORC(4)                                                        PSUDL020
      W=WO/2.4                                                          PSUDL021
      DO 10 JST=1,6                                                     PSUDL022
      SUDY(JST)=SIGCY(JST,W)                                            PSUDL023
   10 CONTINUE                                                          PSUDL024
   20 IF (SORC(5).EQ.DELZO.AND.K.EQ.0) GO TO 40                         PSUDL025
      DELZO=SORC(5)                                                     PSUDL026
      DELZ=DELZO/2.4                                                    PSUDL027
      DO 30 JST=1,6                                                     PSUDL028
      SUDZ(JST)=SIGCZ(JST,DELZ)                                         PSUDL029
   30 CONTINUE                                                          PSUDL030
   40 RETURN                                                            PSUDL031
      END                                                               PSUDL032
```

SUBROUTINE QG3

Purpose:

To call the line function based on a three point Gaussian quadrature procedure.

Input:

1. Wind directions at edges of sector.
2. Name of function to be called.

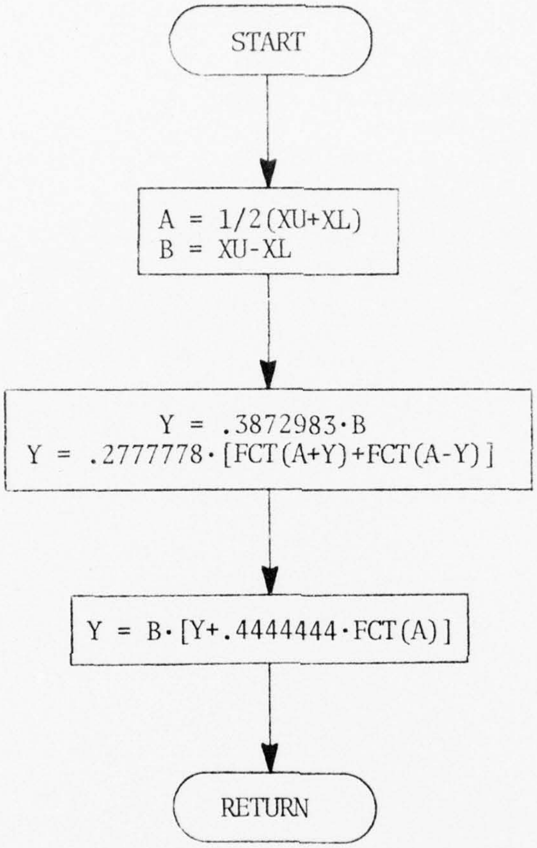
Output:

The line source coupling coefficient.

Procedure:

The three point Gaussian quadrature.

SUBROUTINE QG3(XL,XU,FCT,Y)



```
      SUBROUTINE QG3(XL,XU,FCT,Y)                                       QG300000
C                                                                       QG300001
C     THIS ROUTINE IS A THREE POINT GAUSSIAN QUADRATURE PROCEDURE        QG300002
C                                                                       QG300003
      A=.5*(XU+XL)                                                      QG300004
      B=XU-XL                                                           QG300005
      Y=.3872983*B                                                      QG300006
      Y=.2777778*(FCT(A+Y)+FCT(A-Y))                                    QG300007
      Y=B*(Y+.4444444*FCT(A))                                           QG300008
      RETURN                                                            QG300009
      END                                                               QG300010
```

SUBROUTINE QG4

Purpose:

To call the line function based on a four point Gaussian quadrature procedure.

Input:

1. Wind directions at edges of sector.
2. Name of function to be called.

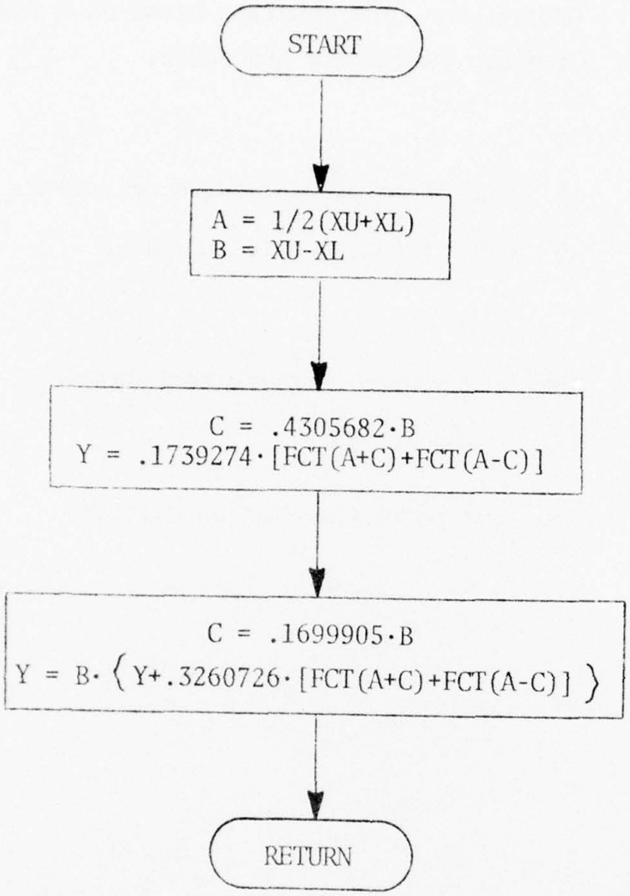
Output:

The line source coupling coefficient.

Procedure:

The four point Gaussian quadrature.

SUBROUTINE QG4(XL,XU,FCT,Y)


START
A = 1/2(XU+XL)
B = XU-XL
C = .4305682·B
Y = .1739274·[FCT(A+C)+FCT(A-C)]
C = .1699905·B
Y = B·{Y+.3260726·[FCT(A+C)+FCT(A-C)]}
RETURN

```
      SUBROUTINE QG4(XL,XU,FCT,Y)
C
C    THIS ROUTINE IS A FOUR POINT GAUSSIAN QUADRATURE PROCEDURE
C
      A=.5*(XU+XL)
      B=XU-XL
      C=.4305682*B
      Y=.1739274*(FCT(A+C)+FCT(A-C))
      C=.1699905*B
      Y=B*(Y+.3260726*(FCT(A+C)+FCT(A-C)))
      RETURN
      END
```

SUBROUTINE QMOD

Purpose:

To compute the linear distribution, in inverse length, of the pollution along a runway due to aircraft emission during landing or takeoff.

Input:

YSI Distance along runway measured from tip of exhaust plume near starting end of runway

TAIL Length or penetration of exhaust plume of aircraft at rest

DL Length of smoke slug on runway

A Acceleration (or deceleration) of aircraft

V12 Initial velocity squared

VS Average velocity of exhaust particles relative to air mass in exhaust plume

WS2 Wind speed squared

WSC 2·wind speed·(- cosine of angle between runway and wind vector)

RR A/G, where A is acceleration and G is the normalization constant for line density

Output:

QL The linear distribution of pollution

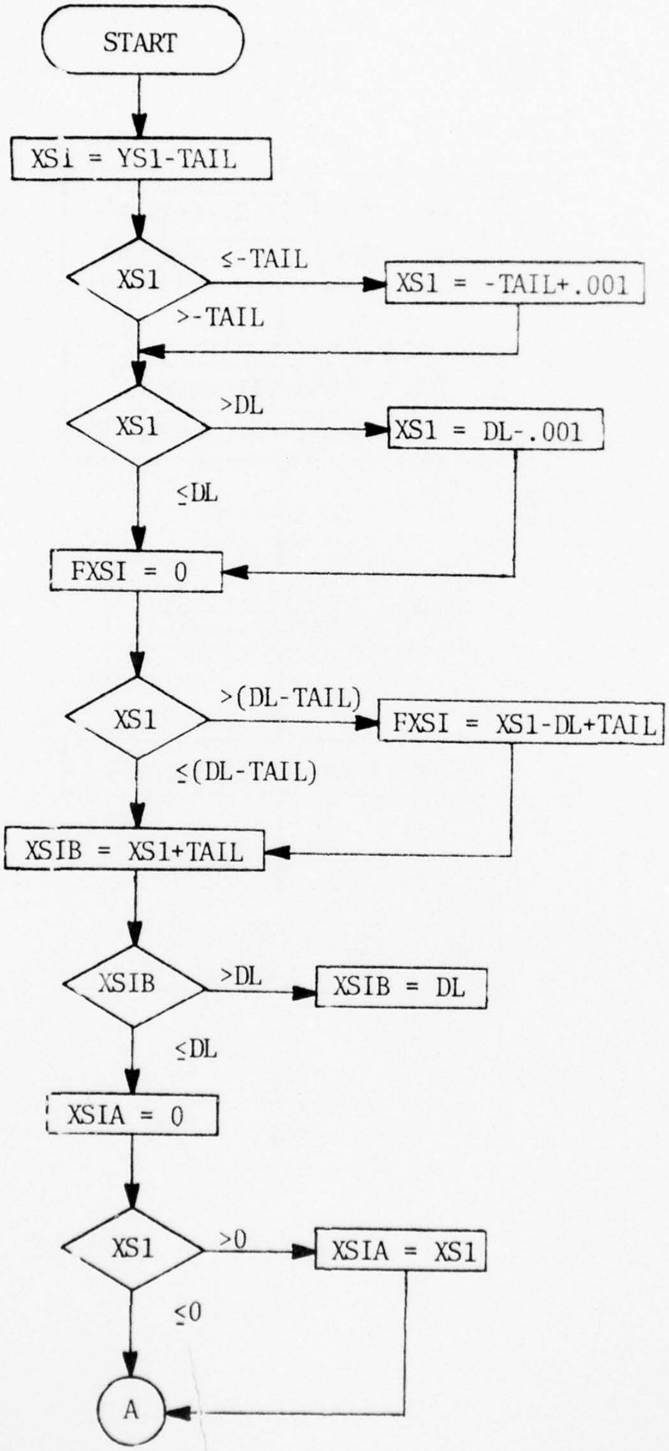
Procedure:

1. Convert the quantity YSI to XSI, the distance measured from the physical end of the runway.
2. Use the line density formula to compute QL in inverse length.

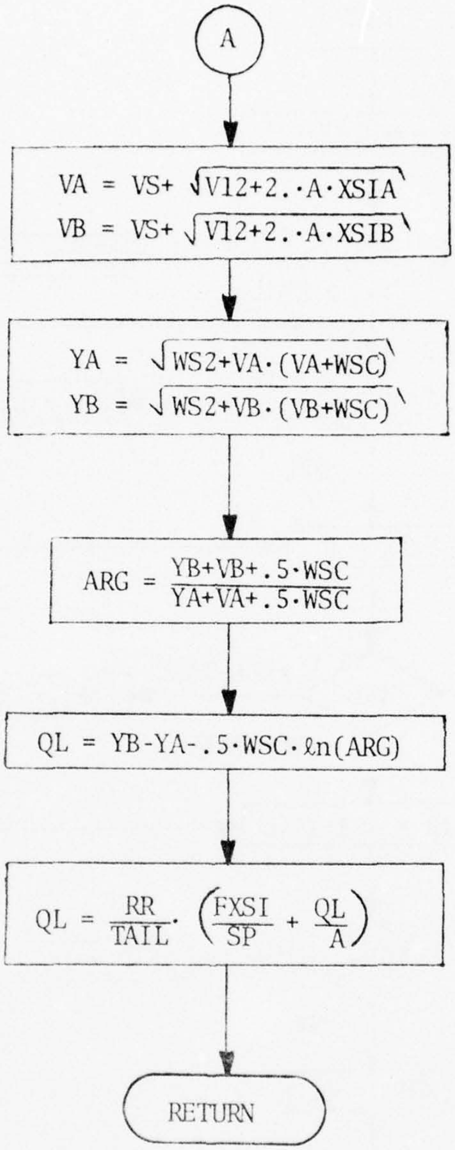
Subroutines Called:

None

SUBROUTINE QMOD



SUBROUTINE QMOD (Contd.)


A
$VA = VS+\sqrt{V12+2.\cdot A\cdot XSIA}$
$VB = VS+\sqrt{V12+2.\cdot A\cdot XSIB}$
$YA = \sqrt{WS2+VA\cdot(VA+WSC)}$
$YB = \sqrt{WS2+VB\cdot(VB+WSC)}$
$ARG = \frac{YB+VB+.5\cdot WSC}{YA+VA+.5\cdot WSC}$
$QL = YB-YA-.5\cdot WSC\cdot \ell n(ARG)$
$QL = \frac{RR}{TAIL}\cdot\left(\frac{FXSI}{SP}+\frac{QL}{A}\right)$
RETURN

```
      SUBROUTINE QMOD (YS1,QL)
C
C    THIS ROUTINE COMPUTES THE LINEAR DISTRIBUTION, IN INVERSE LENGTH,
C    OF THE POLLUSION ALONG A RUNWAY DUE TO AIRCRAFT EMMISION
C    DURING LANDING OR TAKEOFF
C
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,
     . V1,V2,DL,TIME,EMIS(6),NPOL
      COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,
     . WS2,WSC,RR,SP
      XS1 = YS1 - TAIL
      IF (XS1 .LE. -TAIL) XS1 = -TAIL + .001
      IF (XS1 .GT. DL) XS1 = DL - .001
      FXSI = 0.
      IF (XS1 .GT. (DL-TAIL)) FXSI = XS1 - DL + TAIL
   30 XSIB = XS1 + TAIL
      IF (XSIB .GT. DL) XSIB = DL
      XSIA = 0.
      IF (XS1 .GT. 0) XSIA = XS1
      ROOTB = V12 + 2.*A*XSIB
      ROOTA = V12 + 2.*A*XSIA
      VA = SQRT(ROOTA) + VS
      VB = SQRT(ROOTB) + VS
      YA = SQRT(WS2 + VA *(VA + WSC))
      YB = SQRT(WS2 + VB *(VB + WSC))
      ARG = (YB + VB + WSC/2.) / (YA + VA + WSC/2.)
      QL = YB - YA - WSC/2. * ALOG(ARG)
      QL = RR / TAIL * (FXSI / SP + QL / A)
      RETURN
      END
```

SUBROUTINE READ

Purpose:

1. To read master source tape, thereby prcviding the emission inventory and related data to the source emission distribution subroutines.

2. To set up the wind-dependent sources as random access disk data sets.

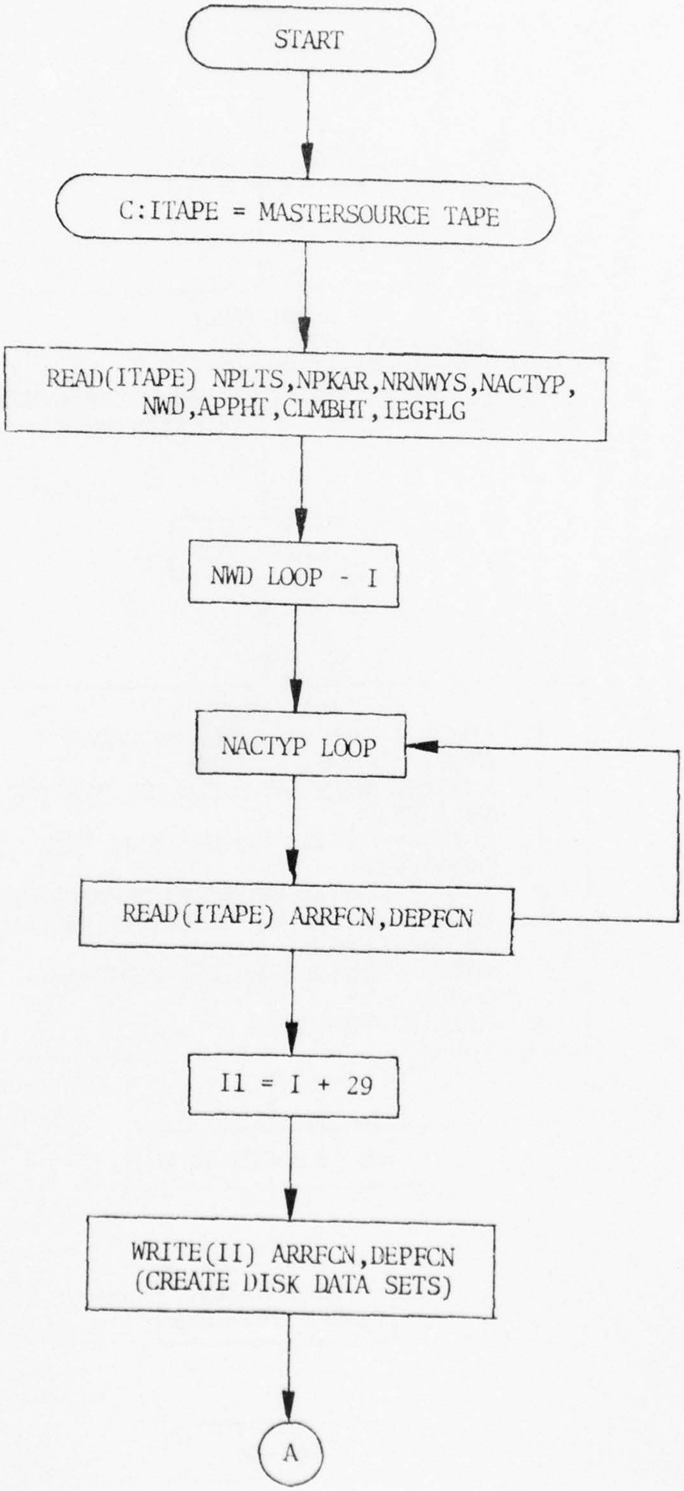
Input:

Master source tape.

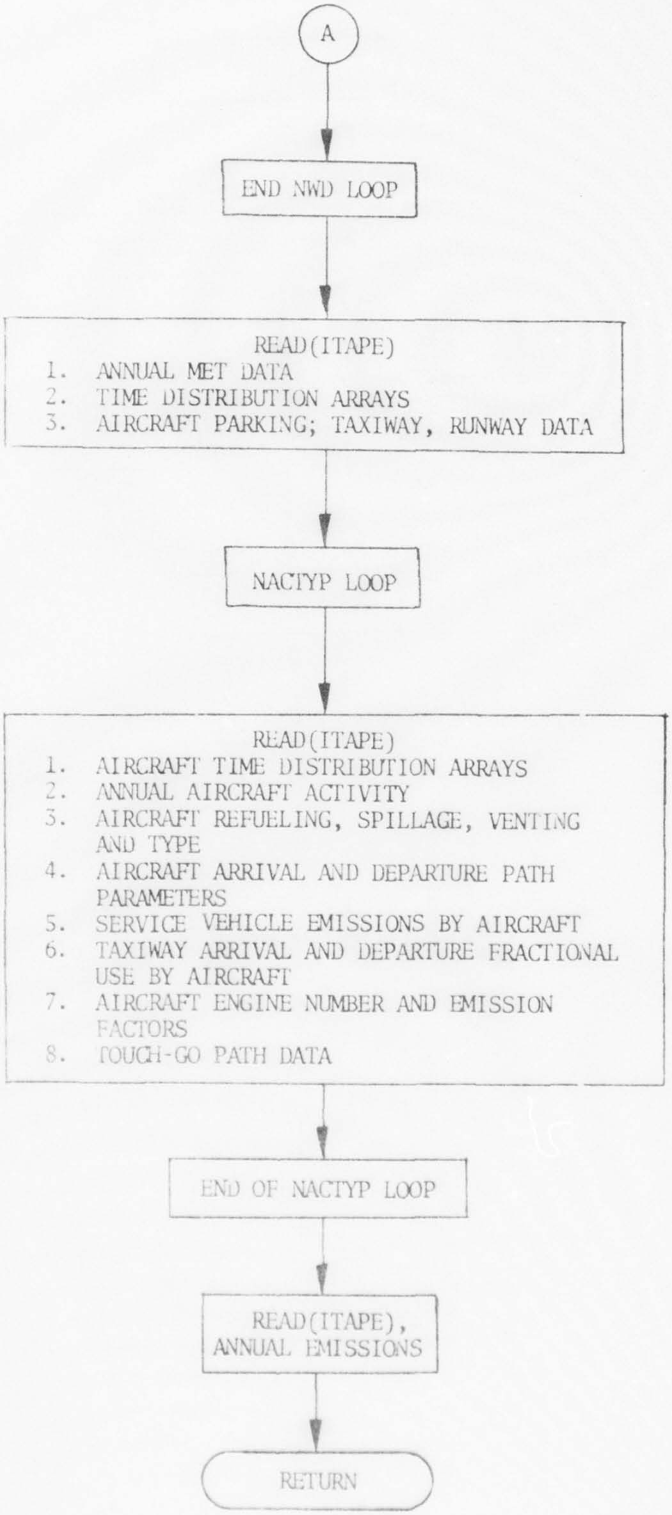
Output:

ARRFCN, DEPFCN to disk.

SUBROUTINE READ



SUBROUTINE READ (Continued)


A
END NWD LOOP
READ(ITAPE)
1. ANNUAL MET DATA
2. TIME DISTRIBUTION ARRAYS
3. AIRCRAFT PARKING; TAXIWAY, RUNWAY DATA
NACTYP LOOP
READ(ITAPE)
1. AIRCRAFT TIME DISTRIBUTION ARRAYS
2. ANNUAL AIRCRAFT ACTIVITY
3. AIRCRAFT REFUELING, SPILLAGE, VENTING AND TYPE
4. AIRCRAFT ARRIVAL AND DEPARTURE PATH PARAMETERS
5. SERVICE VEHICLE EMISSIONS BY AIRCRAFT
6. TAXIWAY ARRIVAL AND DEPARTURE FRACTIONAL USE BY AIRCRAFT
7. AIRCRAFT ENGINE NUMBER AND EMISSION FACTORS
8. TOUCH-GO PATH DATA
END OF NACTYP LOOP
READ(ITAPE), ANNUAL EMISSIONS
RETURN

AD-A047 296 ARGONNE NATIONAL LAB ILL F/G 13/2
AIR QUALITY ASSESSMENT FOR AIR FORCE OPERATIONS - LONG-TERM EMI--ETC(U)
APR 77 D J BINGAMAN

UNCLASSIFIED CEEDO-TR-76-35 NL

3 OF 3
ADA047296

END
DATE
FILMED
1 -78
DDC

```
      SUBROUTINE READ
C
C   THIS ROUTINE READS THE MASTER SOURCE EMISSION TAPE
C
      REAL LNDSPD
      INTEGER  ENGNO
      COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR
      COMMON / RECPT / MRECPT,MAXFIL
      COMMON / DEFALT /   ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FIDENS(7)
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)
      COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),\BLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,25C)
      COMMON /DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
C
      READ   (ITAPE) NPLTS,NPKAR,NRNWYS,NACTYP,NWD,APPHT,CLMBHT,IEGFLG
     . , NLSEGS
      REWIND 30
      DO 2 I=1,NWD
      DO 5 J=1,NACTYP
      READ (ITAPE) ((ARRFCN(L,J,K),DEPFCN(L,J,K),L=1,24),K=1,6)
    5 CONTINUE
      WRITE(30)ARRFCN,DEPFCN
    2 CONTINUE
      REWIND 30
      MRECPT=1
      MAXFIL=NWD
      READ (ITAPE) (JES1(I),I=1,NACTYP)
      READ (ITAPE) TBAR,ADD,PA,WSBAR,DTBAR
      READ   (ITAPE) VHMLMO,VHML ≤,VHMLqr└ceaBmO└ceaAAh,c aA¢r└c enM └T
     . CVENDY,CVENHR,FLMO,FLDY,FLHR
      READ   (ITAPE) NIBTT,NIBSEG,IIBSEG,NOBTT,NOBSEG,IOBSEG
      READ   (ITAPE) IDOBTW,IDIBTW,IDPRKA,PAREA,IDIBPA,IDOBPA,NPASQ
      READ   (ITAPE) RNWY,IUSWD,DISRNW
      READ (ITAPE) ((ACLNSG(II,JJ),II=1,12),JJ=1,NLSEGS)
      DO 40 J=1,NACTYP
      READ(ITAPE) (ACMO(K,J),K=1,13),(ACDY(K,J),K=1,2),(ACHR(K,J),K=1,24)
      READ   (ITAPE) ANNARR(J),ANNDEP(J),ANNTGO(J),ACFUEL(J),ARFLVT(J),
     . DPFLVT(J),ACSPIL(J),IACTYP(J)
      READ (ITAPE) DSCNT1(J),DSCNT2(J),ASCNT1(J),ASCNT2(J),
     . TXISPD(J),LNDSPD(J),APSPD1(J),APSPD2(J),TOSPD(J),COSPD1(J),
     . COSPD2(J),SRTUPT(J),EGCHKT(J),SHTDNT(J),TOWT(J),APPHT2(J),
     . COHT1(J),IDRR(J)
      READ (ITAPE) ((ARSVEM(K,J,L),DPSVEM(K,J,L),L=1,5),K=1,6),
     . ((TTARFR(K,J,L),TTDPFR(K,J,L),K=1,8),L=1,6)
      READ (ITAPE) (ENGNO(J,L),L=1,2),((ACEMFC(J,K,L),K=1,10),L=1,6)
      READ   (ITAPE) ((TGO(K,L,J),K=1,3),L=1,4)
   40 CONTINUE
    4 READ(ITAPE,END=3)
      GO TO 4
    3 CONTINUE                                                      READ0062
      RETURN                                                        READ0063
      END                                                           READ0064
```

FUNCTION RISE

Purpose:

To calculate the plume rise using either the Carson-Moses or Holland plume rise formula.

Input:

Stack parameters, current wind speed and stability, temperature, and the plume rise flag.

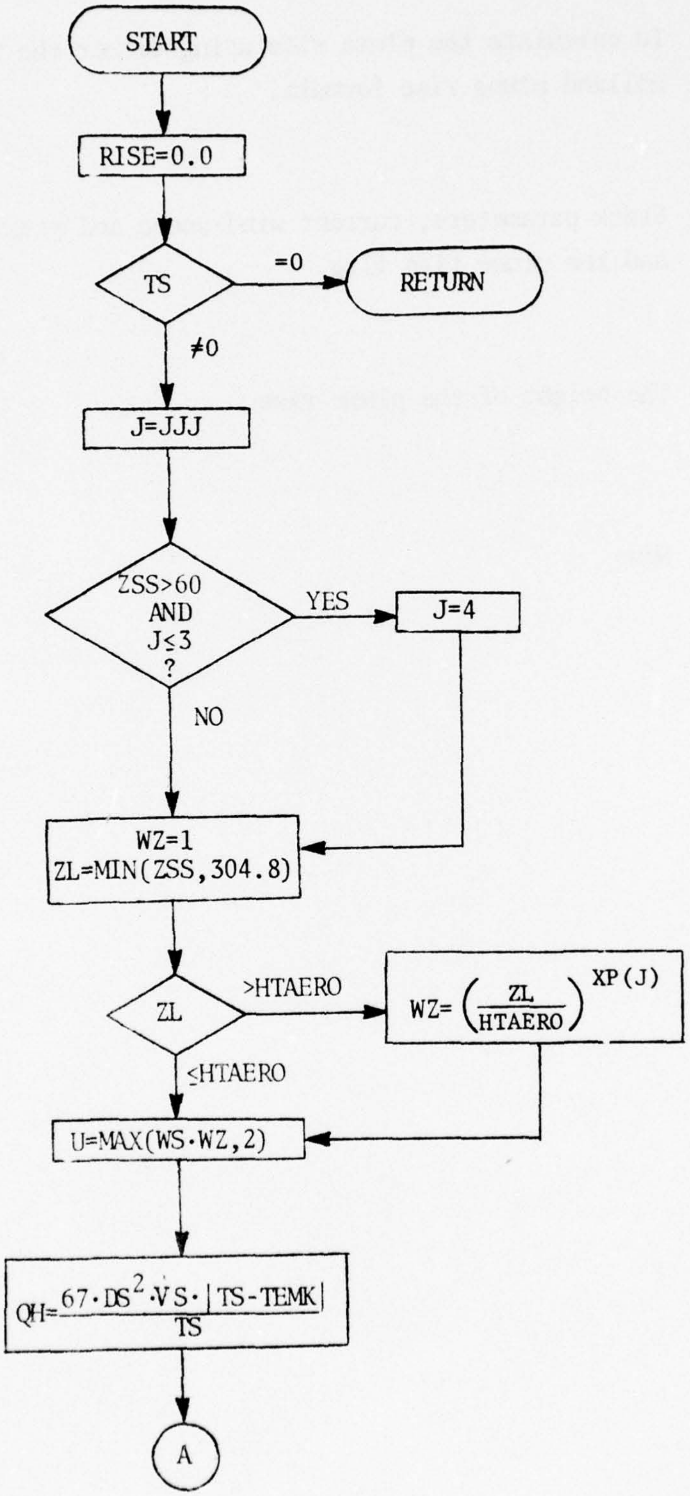
Output:

The height of the plume rise.

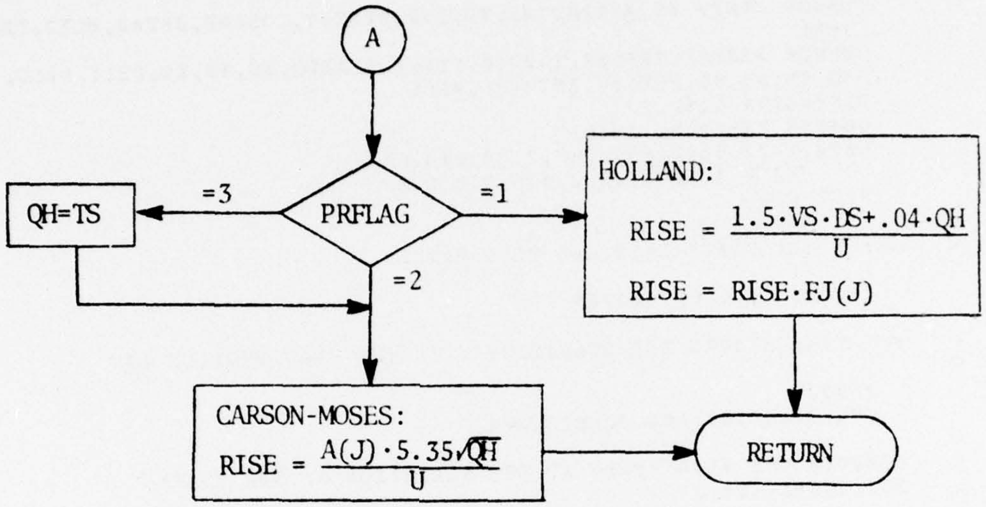
Subroutines Called:

None

FUNCTION RISE



FUNCTION RISE (Cont'd.)


A
PRFLAG
=3
=1
=2
QH=TS
HOLLAND:
RISE = $\frac{1.5 \cdot VS \cdot DS + .04 \cdot QH}{U}$
RISE = RISE·FJ(J)
CARSON-MOSES:
RISE = $\frac{A(J) \cdot 5.35\sqrt{QH}}{U}$
RETURN

```
      FUNCTION RISE(ZSS,JJJ)
C
C     THIS FUNCTION CALCULATES THE PLUME RISE
C        ZSS IS THE PHYSICAL STACK HEIGHT MODIFIED FOR DOWNWASH
C            EFFECTS, IF ANY
C        JJJ IS THE AMBIENT STABILITY
C
      COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
     . TEMK
      COMMON /INFO/ IRECEP,IWNDIR,ITYPE,HTAERO,XS,YS,ZS,DELY,DELZ,
     . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      DIMENSION A(6),FJ(6)
      COMMON /WNDPRO/ XP(6)
      DATA A /2.65,2.65,2.65,1.08,2*0.68/,
     .     FJ / 1.2,1.2,1.2,1.0,0.8,0.8/
      RISE=0.0
C
C     CHECK THE STACK EXIT GAS TEMPERATURE
C
      IF (TS.EQ.0.0) RETURN
C
C     FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW
C
      J=JJJ
      IF(ZSS.GT.60.AND.J.LE.3)J=4
C
C     COMPUTE THE WIND SPEED AT THE ELEVATION OF THE STACK
C     FOR STABILITY J
C
      WZ=1.0
      ZL=AMIN1(ZSS,304.8)
      IF(ZL.GT.HTAERO) WZ=(ZL/HTAERO)**XP(J)
      U=AMAX1(WS*WZ,2.0)
C
C     COMPUTE THE THERMAL EMISSION RATE
C
      QH=67.0*DS*DS*VS*ABS(TS-TEMK)/TS
      IF (PRFLAG.EQ.1.0) GO TO 1
      IF (PRFLAG.EQ.3.0) QH=TS
C
C     CARSON-MOSES PLUME RISE FORMULA
C
      RISE=A(J)*5.35*SQRT(QH)/U
      RETURN
C
C     HOLLAND PLUME RISE FORMULA
C
    1 CONTINUE
      RISE=1.5*VS*DS/U+0.04*QH/U
      RISE=RISE*FJ(J)
      RETURN
      END
```

FUNCTION RRDIST

Purpose:

To calculate the length of runway necessary for takeoff using aircraft dependent equations.

Input:

Aircraft identification, pressure altitude, ambient temperature and wind speed, and aircraft takeoff weight.

Output:

Takeoff length in feet of runway roll to liftoff.

Procedure:

For a given aircraft, use the proper set of takeoff equations provided by the USAF.

Subroutines Called:

None

```
      FUNCTION RRDIST (IR,PA,T,GW,WS)
C
C     FUNCTION CALCULATES RUNWAY ROLL DISTANCE IN FEET
C     IR IS AIRCRAFT IDENTIFICATION NUMBER
C     PA IS PRESSURE ALTITUDE IN HUNDREDS OF FEET
C     T IS TEMPERATURE IN DEGREES FAHRENHEIT
C     GW IS AC TAKE OFF WEIGHT IN THOUSAND POUNDS
C     WS IS THE WIND SPEED IN KNOTS
C
      FGR=0.0
      IF(IR.EQ.100) GO TO 100
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     123,24,25,26,27,28,29,30,31,32,33,34,35,36,37,100,100,100,100,100,
     2 100,100,100,100,100, 100,100,12),IR
 1    CONTINUE
      GO TO 100
 3    CONTINUE
 2    TOF=-(2.78-8.5714E-4*PA)+(1.82E-2+7.2857E-5*PA)*GW
      GR=(1.184E+1-4.2167E-1*T+1.0E-2*T**2-4.583E-5*T**3)+
     .  (4.194+1.7197E-2*T-9.26018E-4*T**2)*TOF+
     .  (1.0457+8.40E-3*T+2.117E-4*T**2+2.98E-7*T**3)*TOF**2
      FGR=(GR-(1.15E-1+9.0E-3*GR)*WS)*100.
      GO TO 100
 4    CONTINUE
 5    TOF=(1.589+6.883E-3*PA+1.2767E-4*PA**2)+
     .  (6.819E-3+1.1007E-4*PA-3.924E-7*PA**2)*T+
     .  (5.979E-5+3.38096E-7*PA+8.532E-9*PA**2)*T**2
      GR=(-13.25+8.75E-1*GW-1.25E-2*GW**2)+
     .  (1.3925E+1-9.275E-1*GW+2.125E-2*GW**2)*TOF
      FGR=(GR-(1.316E-1+8.748E-3*GR)*WS)*100.
      GO TO 100
 6    TOF=(9.3937E-1+2.0947E-2*PA+2.005E-4*PA**2)+
     .  (3.746467E-2+4.05625E-4*PA)*T+
     .  (1.9928E-4-5.75006E-6*PA+1.40234E-7*PA**2)*T**2
      GR=(1.4307E+1-7.57144E-1*GW+2.6785E-2*GW**2)+
     .  (1.67257E+1-1.17762*GW+2.7381E-2*GW**2)*TOF
      FGR=(GR-(2.412799E-2+7.82971E-3*GR)*WS)*100.
      GO TO 100
 7    TOF=(-1.06E-3+1.674E-2*PA+8.1888E-5*PA**2)+
     .  (1.36E-2+9.592E-6*PA+1.755E-6*PA**2)*T+
     .  (5.1099E-5+1.2899E-6*PA-6.123E-9*PA**2)*T**2
      GR=(-1.423E+1+6.349998E-1*GW+1.6667E-3*GW**2)+
     .  (6.1857-3.2179E-1*GW+8.214E-3*GW**2)*TOF
      FGR=(GR-(6.293E-2+7.328E-3*GR)*WS)*100.
      GO TO 100
 8    TOF=(9.503E-2+3.313E-2*PA+1.3666E-4*PA**2)+
     .  (2.2546E-2+1.7848E-4*PA-4.04E-6*PA**2)*T+
     .  (1.3438E-4-1.2166E-6*PA+4.1854E-8*PA**2)*T**2
      GR=(2.95E+1-2.394*GW+6.497E-2*GW**2)+
     .  (3.1035+7.52E-2*GW-3.186E-3*GW**2)*TOF+
     .  (1.2715-1.5535E-1*GW+4.3889E-3*GW**2)*TOF**2
      FGR=(GR-(-9.0E-2+1.807E-2*GR-7.143E-5*GR**2)*WS)*100.
      GO TO 100
 9    TOF=(3.36455E-3+5.63556E-2*PA)+
     .  (4.417E-2-2.031E-3*PA+5.63E-5*PA**2-3.9954E-7*PA**3)*T+
     .  (-9.2E-5+2.08E-5*PA-5.39E-7*PA**2+3.8E-9*PA**3)*T**2
      GR=(1.65838-3.069E-1*GW+8.1363E-2*GW**2)+
     .  (-3.6111+3.63559E-1*GW)*TOF+
     .  (7.3975E-1-8.78749E-2*GW+3.2487E-3*GW**2)*TOF**2
      FGR=(GR-(5.0E-2+7.4E-3*GR)*WS)*100.
      GO TO 100
 10   TOF=(12.5546-5.7192E-2*PA+1.3075E-4*PA**2)-
```

```
     .  (2.9032E-2-1.0254E-4*PA-1.45125E-7*PA**2)*T                    RRDST062
      GR=((-5.14955E+1+2.57957*GW-1.4425E-2*GW**2)-                    RRDST063
     .  (-1.1535E+1+5.915E-1*GW-4.6828E-3*GW**2)*TOF+                  RRDST064
     .  (-6.2285E-1+3.2375E-2*GW-2.9056E-4*GW**2)*TOF**2)*1000.        RRDST065
      FGR=(3.305E+1+9.729E-1*GR+2.31E-6*GR**2)-                        RRDST066
     .  (8.244+8.3598E-3*GR-1.44E-8*GR**2)*WS                          RRDST067
      GO TO 100                                                        RRDST068
11    TOF=(7.436E-1+4.29E-2*PA)+(2.1276E-2-3.1116E-5*PA)*T             RRDST069
      GR=(1.638E+1-7.78E-1*GW+2.84E-2*GW**2)+                          RRDST070
     .  (3.809-1.947E-1*GW+4.264E-3*GW**2)*TOF+                        RRDST071
     .  (-1.976E-1+1.5737E-2*GW+4.6189E-4*GW**2)*TOF**2                RRDST072
      FGR=(GR-(8.5E-2+8.25E-3*GR)*WS)*100.                             RRDST073
      GO TO 100                                                        RRDST074
12    TOF=(1.1405-4.659E-3*PA+1.28E-5*PA**2)-                          RRDST075
     .  (2.0146E-3-2.46E-5*PA+3.5514E-7*PA**2)*T                       RRDST076
      GR=(-3.0029E+1-9.5225E-2*GW+1.25428E-1*GW**2)-                   RRDST077
     .  (-7.3845E+1+1.20433*GW+1.7857E-1*GW**2)*TOF+                   RRDST078
     .  (-3.57857E+1+7.857E-1*GW+7.14286E-2*GW**2)*TOF**2              RRDST079
      FGR=((3.17413E-1+9.762E-1*GR+2.657E-4*GR**2)-                    RRDST080
     .  (1.1114E-1+7.91177E-3*GR+4.40169E-5*GR**2)*WS)*100.            RRDST081
      GO TO 100                                                        RRDST082
13    TOF=(9.166-5.485E-2*PA)-(3.412E-2-1.8E-4*PA)*T                   RRDST083
      GR=(3.02E+2-3.519E+1*GW+1.841*GW**2)-                            RRDST084
     .  (1.306E+2-1.277E+1*GW+5.4E-1*GW**2)*TOF+                       RRDST085
     .  (2.0687E+1-1.715*GW+6.07E-2*GW**2)*TOF**2-                     RRDST086
     .  (1.1578-8.4228E-2*GW+2.46E-3*GW**2)*TOF**3                     RRDST087
      FGR=(GR-(9.55E-2+7.15E-3*GR)*WS)*100.                            RRDST088
      GO TO 100                                                        RRDST089
14    TOF=(2.336+1.582E-2*PA+1.172E-4*PA**2)+                          RRDST090
     .  (5.604E-3+9.97746E-5*PA-5.8117147E-7*PA**2)*T+                 RRDST091
     .  (9.19269E-5-1.34357E-8*PA+1.61411E-8*PA**2)*T**2               RRDST092
      GR=(7.7366-2.52997E-1*GW+2.385E-3*GW**2)+                        RRDST093
     .  (-2.1071+4.2586E-2*GW+12.748E-4*GW**2)*TOF                     RRDST094
      FGR=(GR-(1.0755E-1+1.4588E-2*GR-7.94156E-5*GR**2)*WS)*100.       RRDST095
      GO TO 100                                                        RRDST096
15    CONTINUE                                                         RRDST097
      GO TO 100                                                        RRDST098
16    TOF=(7.6859-1.15E-1*PA+4.413E-4*PA**2)-                          RRDST099
     .  (2.925E-2-8.1128E-4*PA+6.999E-6*PA**2)*T-                      RRDST100
     .  (2.2289E-4+5.058E-6*PA-7.57E-8*PA**2)*T**2                     RRDST101
      GR=(2.546E+1-2.3388*GW+1.0717E-1*GW**2)-                         RRDST102
     .  (7.9095-6.7434E-1*GW+2.1045E-2*GW**2)*TOF+                     RRDST103
     .  (6.099E-1-5.0858E-2*GW+1.434E-3*GW**2)*TOF**2                  RRDST104
      FGR=(GR-(1.16E-1+7.27E-3*GR-3.64E-6*GR**2)*WS)*100.              RRDST105
      GO TO 100                                                        RRDST106
17    CONTINUE                                                         RRDST107
      GO TO 100                                                        RRDST108
18    TOF=(2.118+1.058E-2*PA+1.014E-4*PA**2)+                          RRDST109
     .  (2.102E-3+1.84E-4*PA-1.177E-6*PA**2)*T+                        RRDST110
     .  (1.001E-4-7.046E-7*PA+1.355E-8*PA**2)*T**2                     RRDST111
      GR=(1.0E-5)+(-1.9687+4.209E-1*GW+3.9445E-2*GW**2)*TOF            RRDST112
      FGR=(GR-(8.363E-2+1.488E-2*GR-9.78E-5*GR**2)*WS)*100.            RRDST113
      GO TO 100                                                        RRDST114
19    TOF=(4.65478+6.94444E-3*T)+(3.257E-1+2.7778E-4*T)*(PA/10.)       RRDST115
      GR=(.1457+3.5625E-2*GW-6.763E-5*GW**2)+                          RRDST116
     .  (5.1428-3.175E-2*GW+7.0089E-5*GW**2)*TOF                       RRDST117
      FGR=(GR-(.1+.0082*GR)*WS)*100.                                   RRDST118
      GO TO 100                                                        RRDST119
20    TOF=(1.2192956+2.2091577E-3*PA+3.380102E-4*PA**2)+               RRDST120
     .  (1.4628966E-2+2.6313968E-4*PA-1.3818053E-7*PA**2)*T-           RRDST121
     .  (2.4891E-4-6.875E-6*PA+7.8125E-8*PA**2)*T**2+                  RRDST122
     .  (2.20314E-6-6.49E-8*PA+7.47E-10*PA**2)*T**3                    RRDST123
```

```
      GR=((2.3806396-5.9265772E-2*GW+6.67969E-4*GW**2)+                 RRDST124
     .  (-1.19933136+5.041098E-2*GW-2.12517E-4*GW**2)*TOF)*10.          RRDST125
      FGR=(1.0+9.7757143E+1*GR+6.4285714E-2*GR**2)-                     RRDST126
     .  (4.6785706+5.4275515E-1*GR+4.438775E-3*GR**2)*WS                RRDST127
      GO TO 100                                                         RRDST128
   21 TOF=(-4.799107E-1 + 3.3165178E-2*PA +2.7902E-4*PA**2) +           RRDST129
     . (2.129E-2 + 2.2538E-4 * PA - 2.9186E-6 * PA ** 2) * T            RRDST130
      GR = (1.16103 + 5.318E-2 * GW + 9.0525E-4 * GW **2 ) +            RRDST131
     . (3.3695E1 - 6.94278E-1 * GW + 3.8559E-3 * GW ** 2 ) * TOF -      RRDST132
     . (-9.041 + 2.307E-1 * GW - 1.264E-3 * GW ** 2 ) * TOF ** 2 +      RRDST133
     . (-1.0706 + 2.477E-2 * GW - 1.108E-4 * GW ** 2 )*TOF ** 3         RRDST134
      FGR=(GR-(2.4131E-1+2.115E-4*GR + 1.935E-4*GR**2)*WS)*100.         RRDST135
      GO TO 100                                                         RRDST136
   22 CONTINUE                                                          RRDST137
   23 TOF=(3.9116E-2+6.3976E-2*PA)+(1.6557E-2-7.6643E-6*PA)*T           RRDST138
      GR=(5.625-9.5E-2*GW+1.3125E-3*GW**2)+                             RRDST139
     .  (8.6496E-1-1.2768E-2*GW+1.077E-4*GW**2)*TOF+                    RRDST140
     .  (4.0067E-1-5.982E-3*GW+3.627E-5*GW**2)*TOF**2                   RRDST141
      FGR=(GR-(1.508E-1+8.625E-3*GR)*WS)*100.                           RRDST142
      GO TO 100                                                         RRDST143
   24 TOF=(5.4067E+1-1.3375E-1*PA-2.2755E-4*PA**2+3.6508E-6*PA**3)-     RRDST144
     .  (7.395E-2-1.71E-4*PA-5.91E-6*PA**2+4.22E-8*PA**3)*T             RRDST145
      GR=(8.6549E+3-7.75196E+1*GW+2.07846E-1*GW**2)-                    RRDST146
     .  (5.6302E+2-4.9948*GW+1.30519E-2*GW**2)*TOF+                     RRDST147
     .  (1.22509E+1-1.07805E-1*GW+2.759985E-4*GW**2)*TOF**2-            RRDST148
     .  (8.8948E-2-7.77463E-4*GW+1.956483E-6*GW**2)*TOF**3              RRDST149
      FGR=(GR-(1.4123219E-1+8.5293578E-3*GR+5.709895E-6*GR**2)*WS)*100. RRDST150
      GO TO 100                                                         RRDST151
   25 TOF=(7.90371+6.68965E-2*PA+2.12622E-4*PA**2)+                     RRDST152
     .  (3.00808E-2+2.67118E-5*PA+9.85E-6*PA**2)*T+                     RRDST153
     .  (1.23149E-4+1.3589E-6*PA-3.1641E-8*PA**2)*T**2                  RRDST154
      GR=(2.1742857+2.04286E-1*GW-1.071429E-2*GW**2)+                   RRDST155
     .  (1.14943-1.2707E-1*GW+5.1785E-3*GW**2)*TOF                      RRDST156
      FGR=(GR-(-2.7327E-2+1.904E-2*GR)*WS+                              RRDST157
     .  (-6.308077E-4+1.94654E-4*GR)*WS**2)*100.                        RRDST158
      GO TO 100                                                         RRDST159
   26 CONTINUE                                                          RRDST160
   27 CONTINUE                                                          RRDST161
   28 CONTINUE                                                          RRDST162
   29 TOF=(7.83935E-1+5.38189E-2*PA)+                                   RRDST163
     .  (1.20408E-2+9.888357E-5*PA-2.32448E-6*PA**2)*T-                 RRDST164
     .  (9.72E-6+1.8278E-6*PA-2.405E-8*PA**2)*T**2                      RRDST165
      GR=(3.18978E+1-1.785*GW+3.602E-2*GW**2)+                          RRDST166
     .  (-8.8285+5.1387E-1*GW-5.679E-3*GW**2)*TOF+                      RRDST167
     .  (-1.76441+4.82709E-2*GW)*TOF**2                                 RRDST168
      FGR=(GR-(8.6457E-2+1.1414E-2*GR)*WS)*100.                         RRDST169
      GO TO 100                                                         RRDST170
   30 TOF=(-2.890514E-1+5.8370956E-2*PA)+                               RRDST171
     .  (4.161561E-2-3.518445E-5*PA)*T+(-6.0515E-5+3.53095E-6*PA)*T**2  RRDST172
      GR=(-2.684337E+1+3.224954*GW)+(-2.0581519+3.7024356E-1*GW)*TOF+   RRDST173
     .  (-8.861357E-1+8.3093188E-2*GW)*TOF**2                           RRDST174
      FGR=(GR-(1.3583333E-1+9.5833E-3*GR)*WS)*100.                      RRDST175
      GO TO 100                                                         RRDST176
   31 TOF=(7.46275E-1+1.789924E-2*PA+1.667729E-4*PA**2)+                RRDST177
     .  (6.1017875E-3+3.4816947E-4*PA-1.6406229E-6*PA**2)*T+            RRDST178
     .  (1.718525E-4-2.621825E-6*PA+4.184375E-8*PA**2)*T**2             RRDST179
      GR=(-7.2378129E+1+3.8485684E+1*GW-6.565*GW**2+3.916E-1*GW**3)+    RRDST180
     .  (-5.477E+1+2.92E+1*GW-4.975*GW**2+2.906E-1*GW**3)*TOF           RRDST181
      FGR=((-1.607758+1.222176*GR-5.64375E-3*GR**2)-                    RRDST182
     .  (.482382E-1+2.2260152E-2*GR-4.7462116E-4*GR**2)*WS)*100.        RRDST183
      GO TO 100                                                         RRDST184
   32 TOF=(1.996+1.69E-2*PA+2.56E-5*PA**2)+                             RRDST185
```

```
     .  (8.64E-3-7.5E-5*PA+1.61E-6*PA**2)*T                            RRDST186
      GR=(6.26E+1-1.299E+1*GW+6.886E-1*GW**2)+                         RRDST187
     .  (-1.0004E+2+2.0317E+1*GW-9.67E-1*GW**2)*TOF+                   RRDST188
     .  (1.30368E+1-2.689*GW+1.403E-1*GW**2)*TOF**2                    RRDST189
      FGR=((-3.3E-1+1.047*GR-8.57E-4*GR**2)-                           RRDST190
     .  (4.22E-2+9.47E-3*GR+1.9898E-5*GR)*WS)*100.                     RRDST191
      GO TO 100                                                        RRDST192
33    TCF=(6.6742857E-1+4.4226786E-2*PA)+                              RRDST193
     .  (1.027143E-2+3.051339E-4*PA)*T+(1.74994E-4+5.023E-7*PA)*T**2   RRDST194
      GR=(-1.3766666E+1+1.679166666*GW)+(-3.55+4.71875E-1*GW)*TOF      RRDST195
      FGR=(GR-(1.516666666E-1+1.008333333E-2*GR)*WS)*100.              RRDST196
      GO TO 100                                                        RRDST197
34    CONTINUE                                                         RRDST198
35    CONTINUE                                                         RRDST199
36    TCF=(-9.2083337E-1+5.9113889E-2*PA)+(2.291666E-2-2.7778E-5*PA)*T RRDST200
      GR=(3.711176E+1-1.640279E+1*GW+2.22809*GW**2)+                   RRDST201
     .  (-2.09922E+1+8.6991796*GW-8.4586E-1*GW**2)*TOF+                RRDST202
     .  (2.248949-9.093486E-1*GW+1.061975E-1*GW**2)*TOF**2             RRDST203
      FGR=(GR-(4.3358E-2+2.196E-2*GR)*WS+                              RRDST204
     .  (8.79209E-4+8.21219E-5*GR)*WS**2)*100.                         RRDST205
      GO TO 100                                                        RRDST206
37    TCF=(-6.46E-1+6.7857E-2*PA+2.723E-4*PA**2)+                      RRDST207
     .  (3.69E-2-2.24E-3*PA+3.49E-5*PA**2)*T+                          RRDST208
     .  (1.07E-4+3.85E-5*PA-4.688E-7*PA**2)*T**2                       RRDST209
      GR=(5.38-1.105*GW+1.14E-1*GW**2)+                                RRDST210
     .  (8.02E-1-2.57E-1*GW+2.4E-2*GW**2)*TOF                          RRDST211
      FGR=(GR-(1.6E-2+2.44E-2*GR-2.128E-4*GR**2)*WS)*100.              RRDST212
      GO TO 100                                                        RRDST213
100   RRDIST=FGR                                                       RRDST214
      RETURN                                                           RRDST215
      END                                                              RRDST216
```

SUBROUTINE SIGTZ

Purpose:

At the first entry to the routine, to compute the critical distances in meters for each wind speed and stability class. At each subsequent entry to compute the vertical dispersion in meters.

Input:

1. At the first entry, mixing heights for all wind speeds and stability classes.
2. At all subsequent entries:
   a. Stability class
   b. Downwind distance

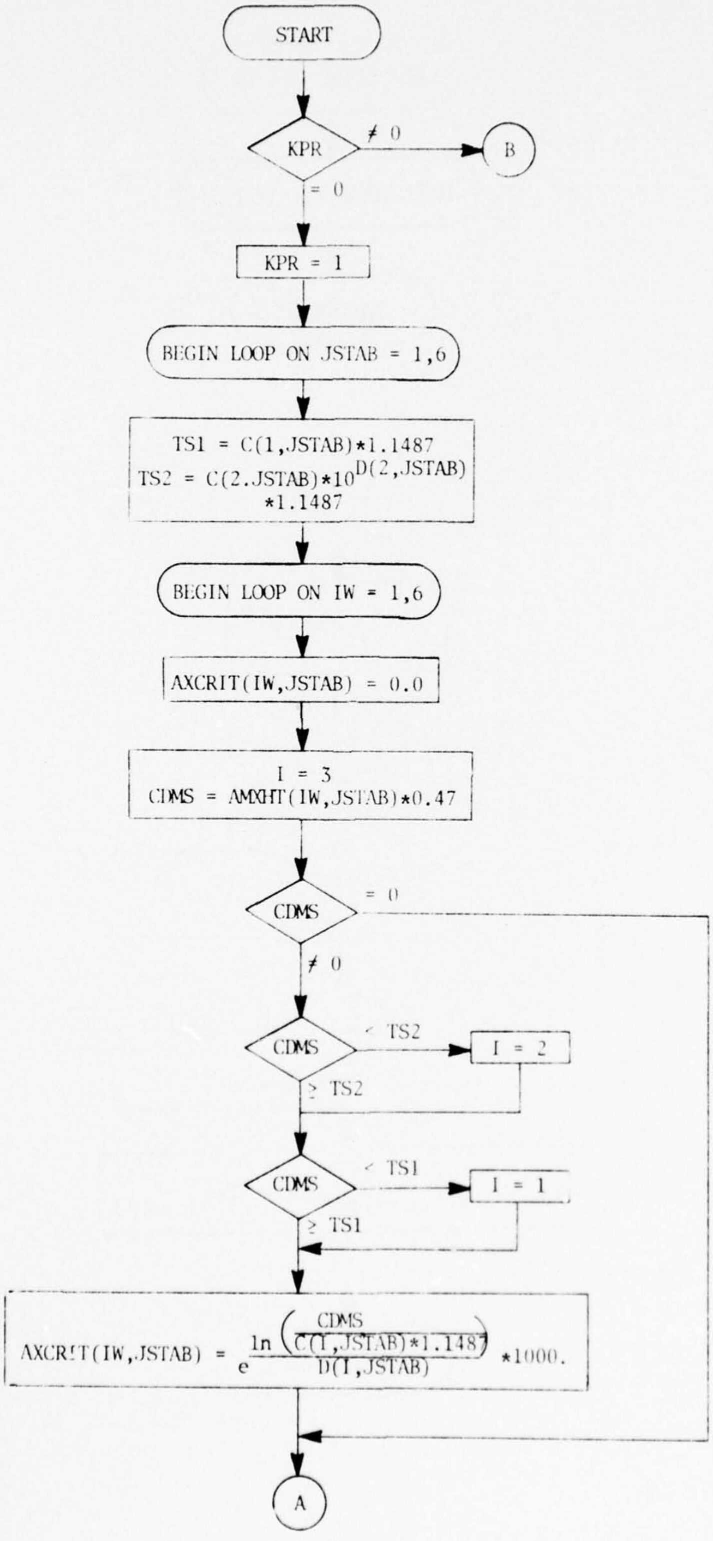
Output:

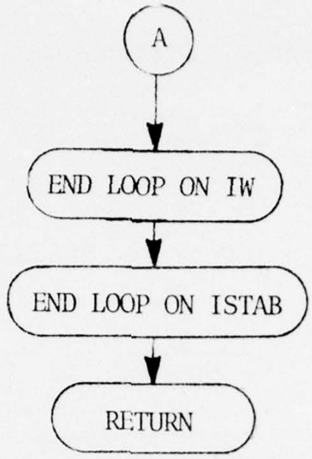
1. At the first entry the critical distance array is filled.
2. At all other entries, the vertical dispersion is returned.

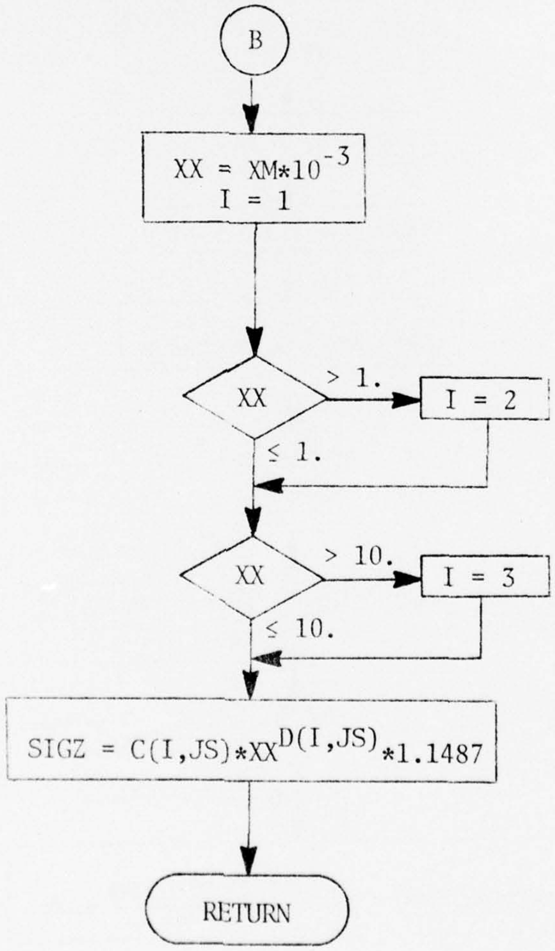
Subroutines Called:

None

SUBROUTINE SIGTZ(JS,XM,SIGZ)


START
KPR
≠ 0
B
= 0
KPR = 1
BEGIN LOOP ON JSTAB = 1,6
TS1 = C(1,JSTAB)*1.1487
TS2 = C(2.JSTAB)*10^D(2,JSTAB) *1.1487
BEGIN LOOP ON IW = 1,6
AXCRIT(IW,JSTAB) = 0.0
I = 3
CDMS = AMXHT(IW,JSTAB)*0.47
CDMS
= 0
≠ 0
CDMS
< TS2
I = 2
≥ TS2
CDMS
< TS1
I = 1
≥ TS1
AXCRIT(IW,JSTAB) = e^(ln(CDMS / C(I,JSTAB)*1.1487) / D(I,JSTAB)) *1000.
A


A
END LOOP ON IW
END LOOP ON ISTAB
RETURN


B
XX = XM*10^-3
I = 1
XX
> 1.
I = 2
≤ 1.
XX
> 10.
I = 3
≤ 10.
SIGZ = C(I,JS)*XX^D(I,JS)*1.1487
RETURN

```
      SUBROUTINE SIGTZ(JS,XM,SIGZ)                                      SIGTZ000
C                                                                       SIGTZ001
C    THIS ROUTINE COMPUTES THE VERTICAL DISPERSION OR                   SIGTZ002
C    DOWNWIND DISTANCE IN METERS                                        SIGTZ003
C                                                                       SIGTZ004
      COMMON /CONS/ PI4,PI8,PI16,KPR,AMXHT(6,6),AXCRIT(6,6)             SIGTZ005
      DIMENSION C(3,6),D(3,6)                                           SIGTZ006
      DATA C /470.,470.,470.,110.,110.,110.,60.,60.,60.,33.,33.,40.,    SIGTZ007
     . 21.5,21.5,36.,14.,14.,23.5/                                      SIGTZ008
      DATA D / 1.67,2.13,2.13,1.0,1.09,1.09,0.92,0.92,0.92,             SIGTZ009
     9 0.80,0.61,0.53,0.70,0.56,0.35,0.78,0.53,0.30/                    SIGTZ010
      IF(KPR.NE.0)GO TO 10                                              SIGTZ011
C                                                                       SIGTZ012
C    AT THIS ENTRY THE CRITICAL DISTANCES FOR EACH WIND SPEED           SIGTZ013
C    AND STABILITY CLASS ARE CALCULATED                                 SIGTZ014
C                                                                       SIGTZ015
      KPR=1                                                             SIGTZ016
      DO 1 JSTAB=1,6                                                    SIGTZ017
      TS1= C(1,JSTAB)*1.1487                                            SIGTZ018
      TS2=C(2,JSTAB)*10.**D(2,JSTAB)*1.1487                             SIGTZ019
      DO 2 IW=1,6                                                       SIGTZ020
      AXCRIT(IW,JSTAB)=0.0                                              SIGTZ021
      I=3                                                               SIGTZ022
      CDMS=AMXHT(IW,JSTAB)*0.47                                         SIGTZ023
      IF(CDMS.EQ.0.0)GO TO 2                                            SIGTZ024
      IF(CDMS      .LT.TS2)I=2                                          SIGTZ025
      IF(CDMS      .LT.TS1)I=1                                          SIGTZ026
      AXCRIT(IW,JSTAB)=EXP(ALOG(CDMS/(C(I,JSTAB)*1.1487))/D(I,JSTAB))   SIGTZ027
     . *1000.                                                           SIGTZ028
    2 CONTINUE                                                          SIGTZ029
    1 CONTINUE                                                          SIGTZ030
      RETURN                                                            SIGTZ031
C                                                                       SIGTZ032
C    AT THIS ENTRY THE VERTICAL DISPERSION IS CALCULATED                SIGTZ033
C                                                                       SIGTZ034
   10 CONTINUE                                                          SIGTZ035
      XX=XM*1.0E-3                                                      SIGTZ036
      I=1                                                               SIGTZ037
      IF(XX.GT.1.)I=2                                                   SIGTZ038
      IF(XX.GT.10.0)I=3                                                 SIGTZ039
      SIGZ=(C(I,JS)*XX**D(I,JS))*1.1487                                 SIGTZ040
      RETURN                                                            SIGTZ041
      END                                                               SIGTZ042
```

FUNCTION SIGY
(ENTRY: SIGCY)

Purpose:

To compute the horizontal dispersion coefficient in meters, or at entry SIGCY, to compute the virtual distance corresponding to the initial horizontal dispersion.

Input:

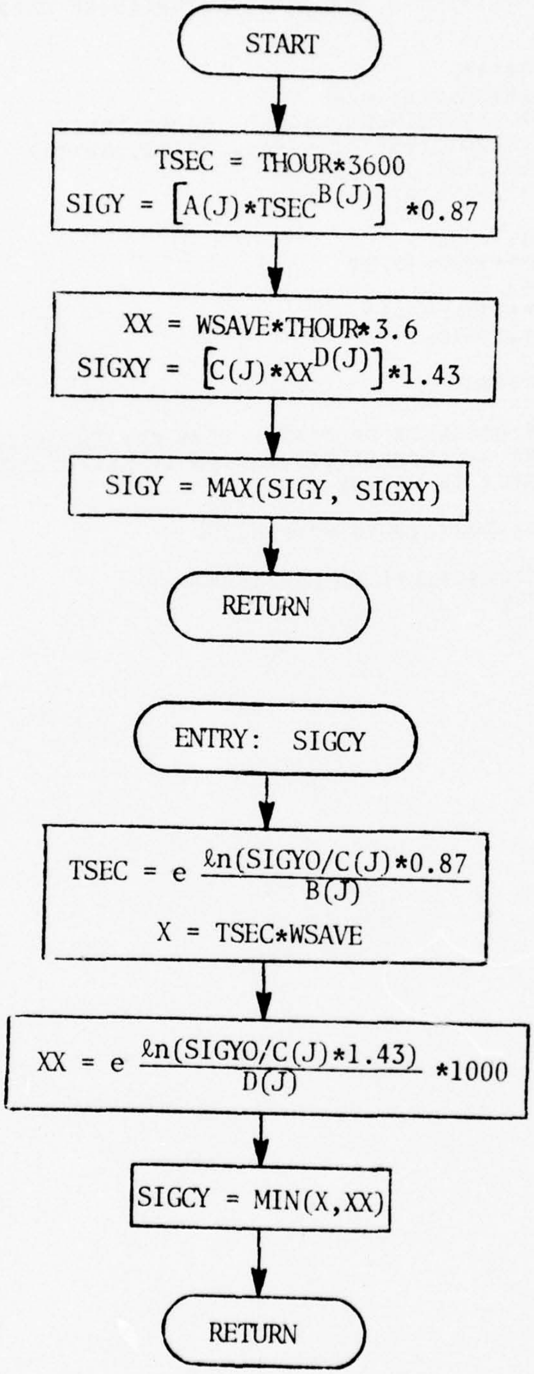
1. Entry SIGY - time of travel in hours
2. Entry SIGCY - horizontal dispersion in meters
3. Stability class and wind speed

Output:

1. SIGY = horizontal dispersion in meters
2. SIGCY = virtual distance in meters

FUNCTION SIGY

ENTRY: SIGCY


START
TSEC = THOUR*3600
SIGY = $\left[A(J)*TSEC^{B(J)}\right]*0.87$
XX = WSAVE*THOUR*3.6
SIGXY = $\left[C(J)*XX^{D(J)}\right]*1.43$
SIGY = MAX(SIGY, SIGXY)
RETURN
ENTRY: SIGCY
TSEC = $e^{\frac{\ln(SIGYO/C(J)*0.87}{B(J)}}$
X = TSEC*WSAVE
XX = $e^{\frac{\ln(SIGYO/C(J)*1.43)}{D(J)}}*1000$
SIGCY = MIN(X,XX)
RETURN

```
      FUNCTION SIGY(J,THOUR)                                            SIGY0000
C                                                                       SIGY0001
C   THIS FUNCTION COMPUTES THE HORIZONTAL DISPERSION COEFFICIENT        SIGY0002
C   IN METERS                                                           SIGY0003
C                                                                       SIGY0004
      COMMON /WDUN/ WSAVE                                               SIGY0005
      DIMENSION A(6),B(6),C(6),D(6)                                     SIGY0006
      DATA A/2.1511,1.5454,1.0606,.68465,.59366,.59366/                 SIGY0007
      DATA B/.87326,.88261,.89031,.88866,.89138,.89138/                 SIGY0008
      DATA C /212.,155.,100.,68.,50.,34./                               SIGY0009
      DATA D/0.89,0.91,0.92,0.93,0.90,0.93/                             SIGY0010
C                                                                       SIGY0011
      TSEC=THOUR*3600.                                                  SIGY0012
      SIGY=(A(J)*TSEC**B(J))*0.87                                       SIGY0013
      XX=WSAVE*THOUR*3.6                                                SIGY0014
      SIGXY=C(J)*(XX**D(J))*1.43                                        SIGY0015
      SIGY=AMAX1(SIGY,SIGXY)                                            SIGY0016
      RETURN                                                            SIGY0017
      ENTRY SIGCY(J,SIGY0)                                              SIGY0018
C                                                                       SIGY0019
C   AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE      SIGY0020
C   INPUT VALUE OF THE HORIZONTAL DISPERSION IS CALCULATED AND          SIGY0021
C   RETURNED AS DISTANCE IN METERS                                      SIGY0022
C                                                                       SIGY0023
      TSEC=EXP(ALOG(SIGY0/(A(J)*0.87))/B(J))                            SIGY0024
      X=TSEC*WSAVE                                                      SIGY0025
      XX=EXP(ALOG(SIGY0/(C(J)*1.43))/D(J))*1000.                        SIGY0026
      SIGCY=AMIN1(X,XX)                                                 SIGY0027
      RETURN                                                            SIGY0028
      END                                                               SIGY0029
```

FUNCTION SIGZ
(ENTRY: SIGCZ)

Purpose:

To compute the vertical dispersion coefficient in meters, or at entry SIGCZ, to compute the virtual distance corresponding to the initial vertical dispersion.

Input:

1. Entry SIGZ - time of travel in hours

2. Entry SIGCZ - vertical dispersion in meters

3. Stability class and wind speed

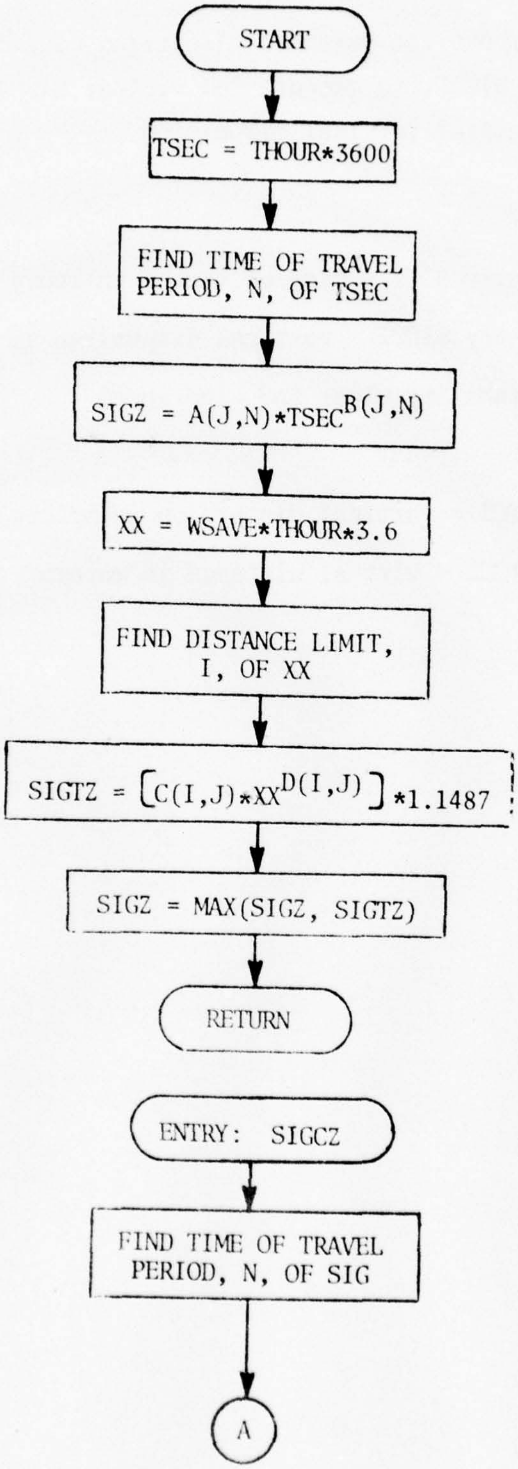
Output:

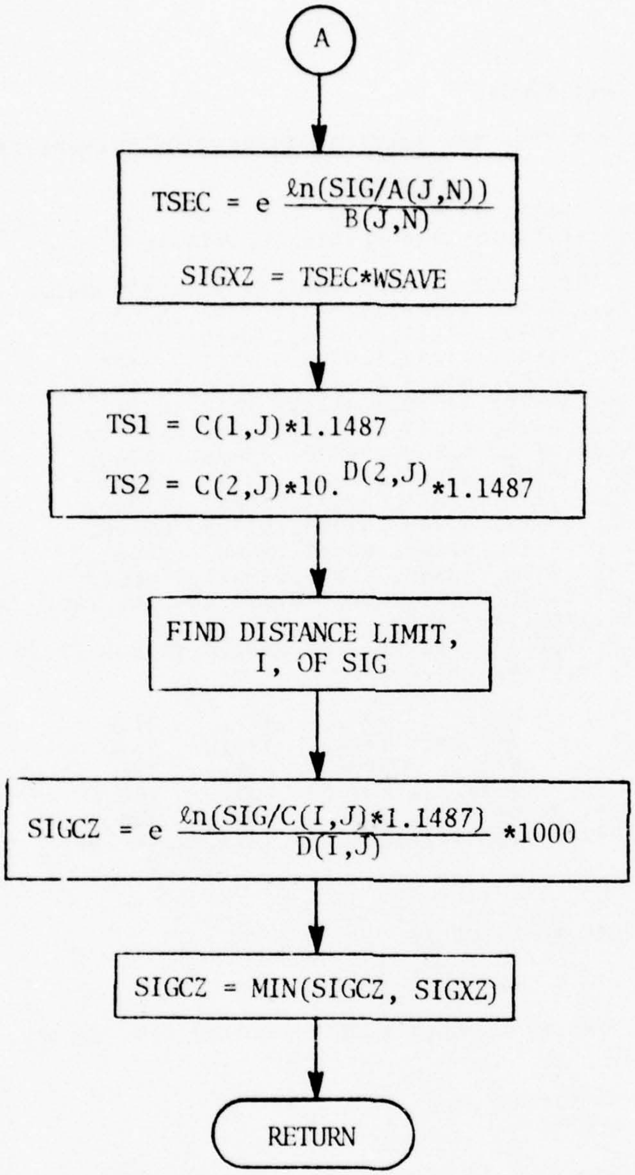
1. SIGZ = vertical dispersion in meters

2. SIGCZ = virtual distance in meters

FUNCTION SIGZ

ENTRY: SIGCZ


START
TSEC = THOUR*3600
FIND TIME OF TRAVEL PERIOD, N, OF TSEC
SIGZ = A(J,N)*TSEC$^{B(J,N)}$
XX = WSAVE*THOUR*3.6
FIND DISTANCE LIMIT, I, OF XX
SIGTZ = $\left[C(I,J)*XX^{D(I,J)}\right]$*1.1487
SIGZ = MAX(SIGZ, SIGTZ)
RETURN
ENTRY: SIGCZ
FIND TIME OF TRAVEL PERIOD, N, OF SIG
A


A
$TSEC = e^{\frac{\ln(SIG/A(J,N))}{B(J,N)}}$
SIGXZ = TSEC*WSAVE
TS1 = C(1,J)*1.1487
$TS2 = C(2,J)*10.^{D(2,J)}*1.1487$
FIND DISTANCE LIMIT, I, OF SIG
$SIGCZ = e^{\frac{\ln(SIG/C(I,J)*1.1487)}{D(I,J)}}*1000$
SIGCZ = MIN(SIGCZ, SIGXZ)
RETURN

```
      FUNCTION SIGZ(J,THOUR)
C
C    THIS FUNCTION COMPUTES THE VERTICAL DISPERSION COEFFICIENT
C    IN METERS
C
       COMMON /WDUN/ WSAVE
       DIMENSION C(3,6),D(3,6),A(6,6),B(6,6),CK(6,6)
       DIMENSION TIME(6)
       DATA TIME/    300.,1000.,3000.,10000.,30000.,172000./
       DATA A/.17122,.27668,.41219,.51921,.50963,.47639,
     1        .11062,.39953,.41219,.57145,.76485,.71936,
     2        .01338,.16640,.41219,1.0813,1.9467,2.3901,
     3        .01338,.16640,.41219,2.2830,2.9850,3.8684,
     3        .01338,.16640,.41219,2.3333,5.7990,16.897,
     3        .01338,.16640,.41219,5.6801,14.599,64.577/
       DATA B/1.2098,1.0572,.92365,.84130,.79689,.76308,
     1        1.2864,.99275,.92365,.82449,.72571,.69082,
     2        1.5922,1.1195,.92365,.73217,.59047,.51700,
     3        1.5922,1.1195,.92365,.63883,.53708,.45686,
     4        1.5922,1.1195,.92365,.63646,.46497,.29621,
     5        1.5922,1.1195,.92365,.55016,.37541,.16667/
       DATA C/470.,470.,470.,110.,110.,110.,60.,60.,60.,33.,33.,40.,
     . 21.5,21.5,36.,14.,14.,23.5/
       DATA D/1.67,2.13,2.13,1.,1.09,1.09,0.92,0.92,0.92,0.80,0.61,0.53,
     . 0.70,0.56,0.35,0.78,0.53,0.30/
       DATA CK/
     1 170.,      115.,      80.,       63.,      48.,      37.,
     2 800.,      380.,      243.25,  170.,     115.,      85.,
     3 4600.,     1300.,     671.,     380.,     220.,      150.,
     4 31279.,    5002.,     2040.32, 820.,     420.,      260.,
     5 179855.2,17111.38,5628.47, 1650.,   700.,      358.,
     6 2900444.,120872.5,28241.86,4312.55,1348.32,481.58/
C
       TSEC=THOUR*3600.
       DO 10 N=1,6
       IF(TSEC.LE.TIME(N))GO TO 20
    10 CONTINUE
       N=6
C
C    TIME OF TRAVEL SHOULD BE LESS THAN 172000 SEC. OR APPROX. 2 DAYS
C
    20 CONTINUE
       SIGZ=(A(J,N)*TSEC**B(J,N))
       XX=WSAVE*THOUR*3.6
       I=1
       IF(XX.GT.1.)  I=2
       IF(XX.GT.10.)  I=3
C
C    CONVERTS FROM A 10 TO 20 MIN. SAMPLING TIME
C    1.1487 = 2**.2, THE 1/5 POWER LAW ONLY APPLIES UP TO 20 MIN.
C    SAMPLING TIMES
C
       SIGTZ=(C(I,J )*XX**D(I,J ))*1.1487
       SIGZ=AMAX1(SIGZ,SIGTZ)
       RETURN
       ENTRY SIGCZ(J,SIG)
C
C    AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORESPONDING TO THE
C    INPUT VALUE OF THE VERTICAL DISPERSION IS CALCULATED AND
C    RETURNED AS DISTANCE IN METERS
C
       DO 110 N=1,6
      IF (SIG.LE.CK(J,N)) GO TO 120                                     SIGZ0062
  110 CONTINUE                                                          SIGZ0063
      N=6                                                               SIGZ0064
  120 CONTINUE                                                          SIGZ0065
      TSEC=EXP(ALOG(SIG/A(J,N))/B(J,N))                                 SIGZ0066
      SIGXZ=TSEC*USAVE                                                  SIGZ0067
      TS1=C(1,J)*1.1487                                                 SIGZ0068
      TS2=C(2,J)*10.**D(2,J)*1.1487                                     SIGZ0069
      I=3                                                               SIGZ0070
      IF (SIG.LT.TS2) I=2                                               SIGZ0071
      IF (SIG.LT.TS1) I=1                                               SIGZ0072
      SIGCZ=EXP(ALOG(SIG/(C(I,J)*1.1487))/D(I,J))*1000.                 SIGZ0073
      SIGCZ=AMIN1(SIGCZ,SIGXZ)                                          SIGZ0074
      RETURN                                                            SIGZ0075
      END                                                               SIGZ0076
```

SUBROUTINE SOURCE

Purpose:

To position the master source tape to read the airbase and environ source inventory data and to call the subroutines which compute the emission rates in micrograms per second at the airbase and environ sources.

Input:

JFLAG, a parameter to indicate whether the diurnal distribution used is input, default or the same as previous hour.

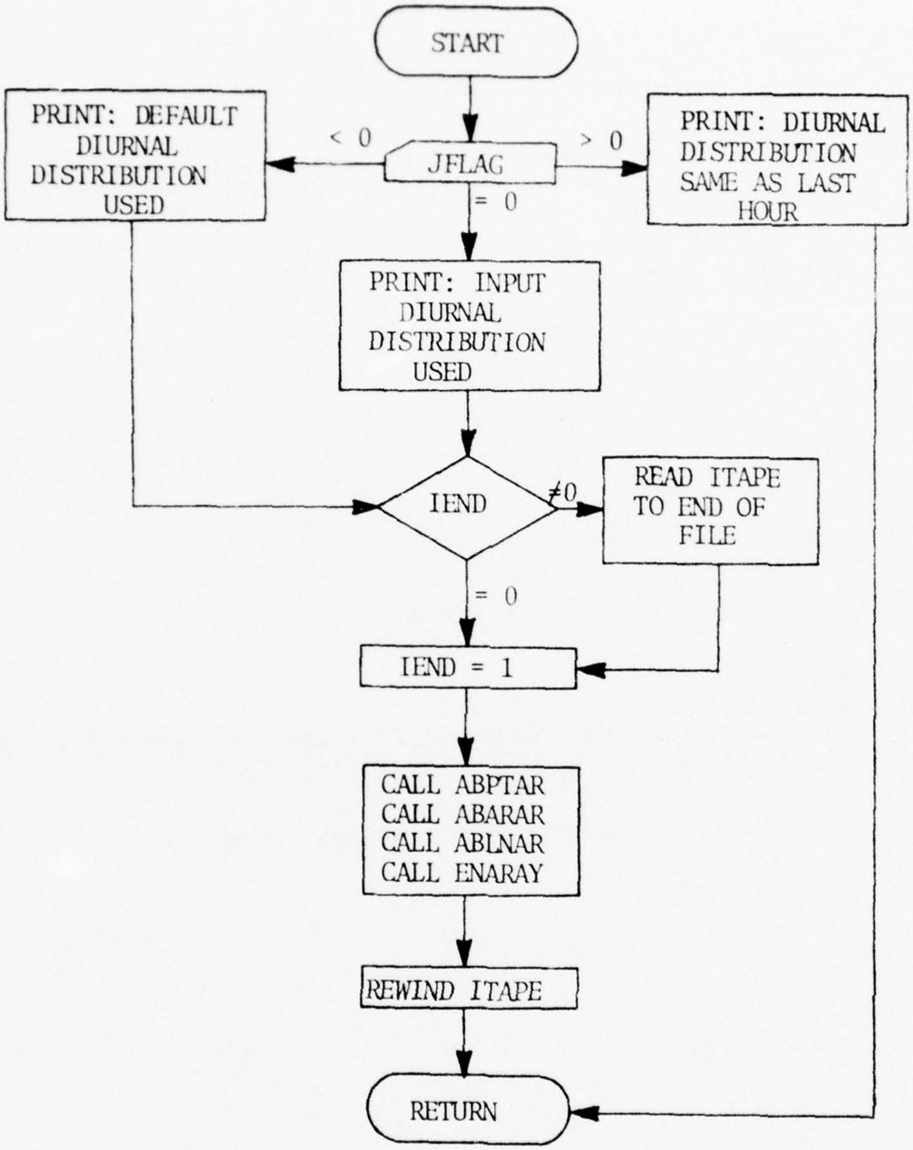
Output:

A statement indicating the diurnal distribution used.

Subroutines Called:

ABPTAR, ABARAR, ABLNAR, ENARAY

SUBROUTINE SOURCE


START
JFLAG
< 0
> 0
= 0
PRINT: DEFAULT DIURNAL DISTRIBUTION USED
PRINT: DIURNAL DISTRIBUTION SAME AS LAST HOUR
PRINT: INPUT DIURNAL DISTRIBUTION USED
IEND
≠0
READ ITAPE TO END OF FILE
= 0
IEND = 1
CALL ABPTAR
CALL ABARAR
CALL ABLNAR
CALL ENARAY
REWIND ITAPE
RETURN

```
      SUBROUTINE SOURCE
C
C     THIS ROUTINE SERVES AS A DRIVER TO CALL SUBROUTINES
C     WHICH COMPUTE THE EMISSION RATES IN MICROGRAMS
C     PER SECOND AT THE AIRBASE AND ENVIRON SOURCES
C
      COMMON / DEFALT / ITAPE
      COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE
      COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)
     . ,LOC1,LOC2,NGEOM,IPT
      DIMENSION NAME(2)
      DATA NAME /4H1/12,4H1    /
      IF (IONCE.EQ.0) GO TO 30
      IEND=1
      IST=1
      GO TO 40
   30 IEND=0
      IONCE=1
      IST=0
   40 CONTINUE
      DAYS=NODAYS
      IF (IST.EQ.1) GO TO 3
    1 FORMAT (I4)
      READ 1, JFLAG
      IF(JFLAG) 8,7,3
    7 PRINT 5
    5 FORMAT(32H0INPUT DIURNAL DISTRIBUTION USED)
      GO TO 4
    8 I=1
      IF (NODAYS.EQ.365) I=2
      PRINT 9,NAME(I)
    9 FORMAT(34H0DEFAULT DIURNAL DISTRIBUTION USED/5X,12HHOUR = 1/24,5X,
     .10HDAY = 1/7,5X,8HMONTH = A4,1H,5X,12HUNIFRC = 0.1)
      GO TO 4
    3 PRINT 6
    6 FORMAT(39H0DIURNAL DISTRIBUTION SAME AS LAST HOUR)
      GO TO 10
    4 IF(IEND.EQ.0) GO TO 12
   11 READ (ITAPE,END=12)
      GO TO 11
   12 IEND=1
      CALL ABPTAR
      CALL ABARAR
      CALL ABLNAR
      CALL ENARAY
   10 REWIND ITAPE
      RETURN
      END
```

SECTION II

INTRODUCTION

After testing of the Long-Term Model showed that computer running times were unacceptably long, it was decided that ANL would develop a simplified version of the Model, termed the Applications Model. This new version had to minimize the loss in accuracy (as compared to the original) of calculated pollutant concentrations and keep the same input requirements.

Since the calculation of dispersion from line sources requires the greatest amount of computer time, it was decided to reduce the large number of aircraft line sources. In the original model, each aircraft type is assigned its own flight path, but in the modified model a single flight path for each runway based on weighted average parameters of all aircraft actually using the runway is established. The numerical value of the emissions by operational mode, remain identical to those in the original version.

The methodology of simplification is summarized as follows:

- Approach, Phase I - This is the longest portion of the arrival path and consists of a straight line segment extending from an altitude of about 1000 meters to an altitude of approximately 300 meters. Computations indicate that dispersion calculations involving these line segments are the most time consuming of all aircraft sources. In addition, previous experience has suggested that air pollution contributions coming from such line sources are negligible compared to other aircraft related sources. Consequently, in the Applications Model, this portion is eliminated.
- Approach, Phase II - This portion of the arrival path consists of a straight line extending from an altitude of approximately 300 meters to the touchdown point on the runway. In the Applications Model, this portion is represented by one line source per runway with the spatial location determined by taking a weighted average of approach parameters of those aircraft using the runway for landings and touch-go operations. Emissions are assumed to be uniformly distributed along the line.

- Landing on runway - One line source per runway is generated and it is assumed that all aircraft touch down 304.8 meters from the end of the runway and proceed to the other end. Emissions are given a non-uniform distribution determined by the initial and final speeds which are taken as a weighted average of those speeds of aircraft actually using the runway for landings.

- Touch-go runway operation - One per runway with the same touchdown point being used as for landing on runway (so that the same approach path can be used for touch-go as for arrivals). Aircraft are on the runway for 304.8 meters prior to lift-off and a uniform emission density is assumed.

- Touch-go departure, Phase I - This operation includes the time from lift-off the runway until the aircraft has climbed to an altitude of approximately 300 meters. One line source per runway is used for this operation with the spatial location of the end points taken as a weighted average of the Phase I departure parameters of those aircraft actually making touch-go operations on the runway. The emissions are assumed to be distributed uniformly.

- Touch-go departure, Phase II - This operation has been eliminated for reasons similar to those of the Phase I approach. It consisted of climbout from approximately 3C0 to 1000 meters.

- The runway roll and climbout phases of a normal aircraft departure operation (as opposed to touch-go climbouts) are handled independently of each other. The climbout phases in the Research Model are essentially analogous to the touch-go climbout except they occur at a different location in space. The Runway Roll is the operation whereby the aircraft accelerates down the runway until gaining sufficient lift to become airborne.

- Runway Roll - If more than one aircraft type uses a runway, two runway roll line sources are defined for each runway unless all the aircraft using the runway have, within ten percent, the same roll distance in which case only one line geometry is defined. In either case, the runway roll line source length(s) is taken to be the weighted average of the roll distances of those aircraft using the runway. Runway roll line sources have non-uniform emission densities determined by the initial and final weighted velocities, i.e., lift-off velocity.

- Climbout, Phase II has been eliminated (see remarks under touch-go departure Phase II).
- Climbout, Phase I consists of either one or two uniform emission density line sources for each runway. If the difference between the maximum and minimum Phase I climbout angles, for aircraft using a runway, is greater than 10° then two distinct limits on line sources are defined. Otherwise, only one is used. If two sources are generated, the class decision criterion is whether or not the aircraft climbout Phase I angle lies in the lower or upper half of the range of climbout angles for that runway.

Comparisons made by running both versions with representative near and far receptors showed clearly that the differences in calculated concentrations were very small.

Table 2 contains a list of the three routines modified or added to create the Applications Model. A brief description is given but a more detailed account of each routine, together with flow charts and computer code listings are given on the following pages. Figure 2 presents the schematic flow diagram of the entire Applications Model.

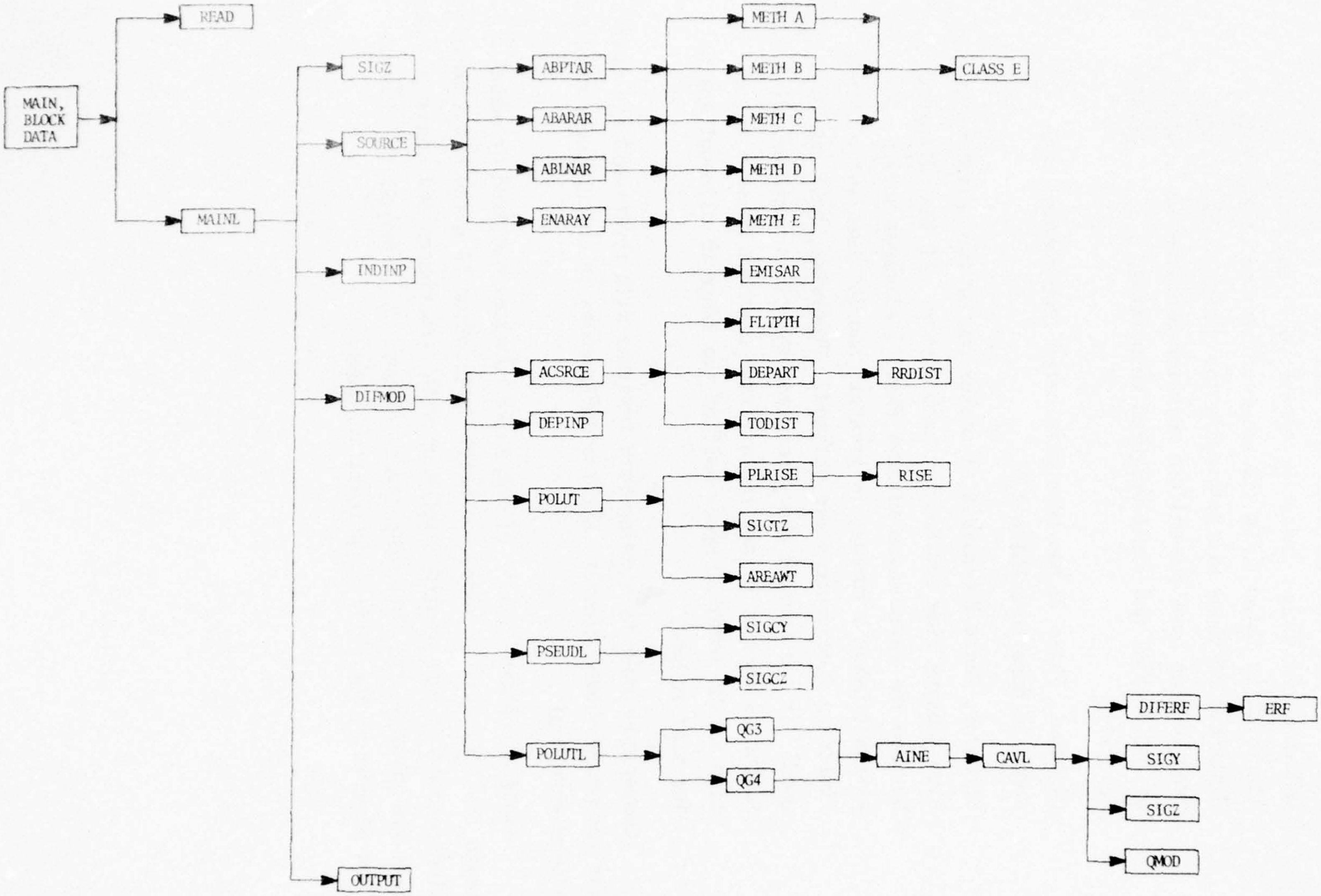


Figure 2. Schematic Flow Diagram of Long Term Applications Model

TABLE 2. LIST OF SUBROUTINES MODIFIED OR ADDED TO THE LONG-TERM EMISSION/DISPERSION MODEL TO CREATE THE APPLICATION MODEL

SUBROUTINE	DESCRIPTION
ASCRCE	Set up the aircraft source arrays and allocate emissions to areas and/or lines.
FLTPTH	Set up the flight path parameters for each runway.
TODIST	Set up the takeoff roll distances on each runway.

SUBROUTINE ACSRCE
(Applications Model)

Purpose:

To set up the aircraft source arrays to be used by the dispersion routines for calculating ground level concentrations. In this version, a single flight path for each runway is established based on weighted average parameters of all the aircraft actually using the runway.

Input:

Basic aircraft data, airbase activity data, points in arrival-departure paths and in training flight paths, meteorological conditions, and the time period of the calculation.

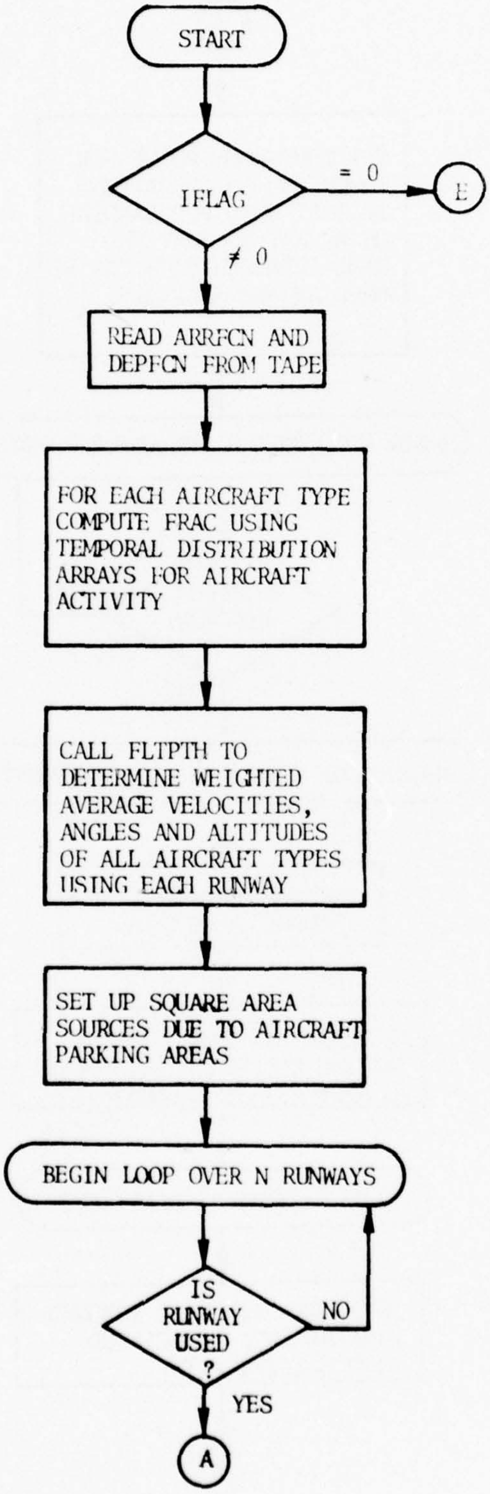
Output:

The arrays ACPT, ACLN and ACAR will contain all source information necessary to calculate dispersion and pollutant concentrations.

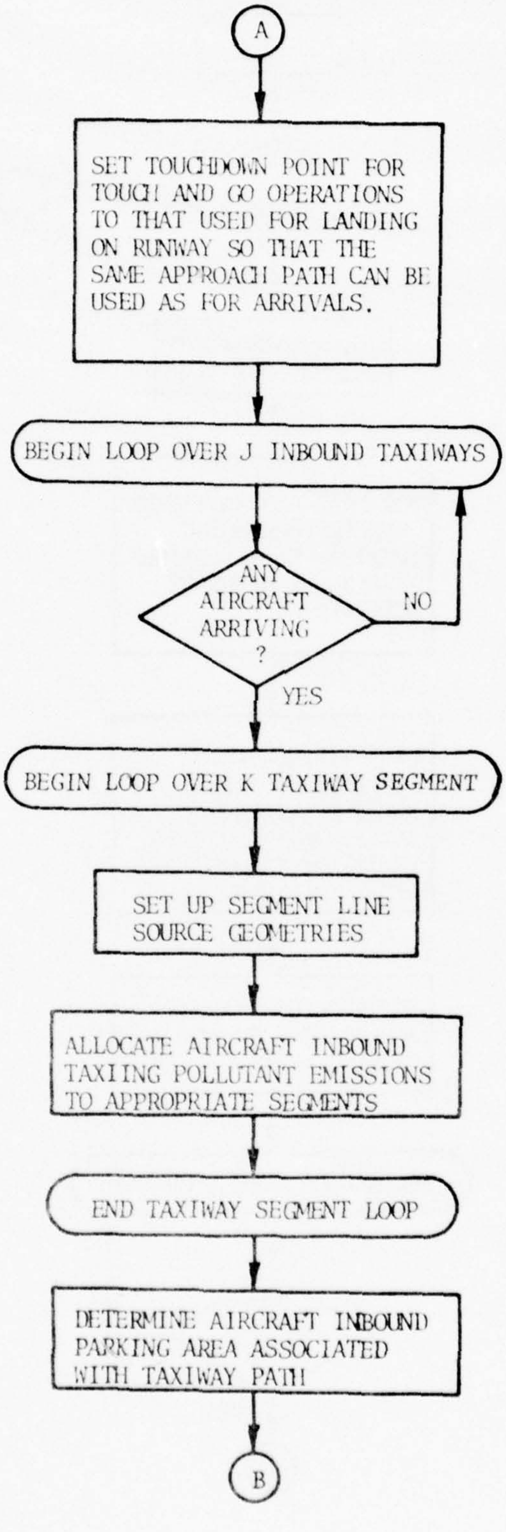
Subroutines Called:

DEPART, FLTPTH, TODIST

SUBROUTINE ACSRCE
(APPLICATIONS MODEL)

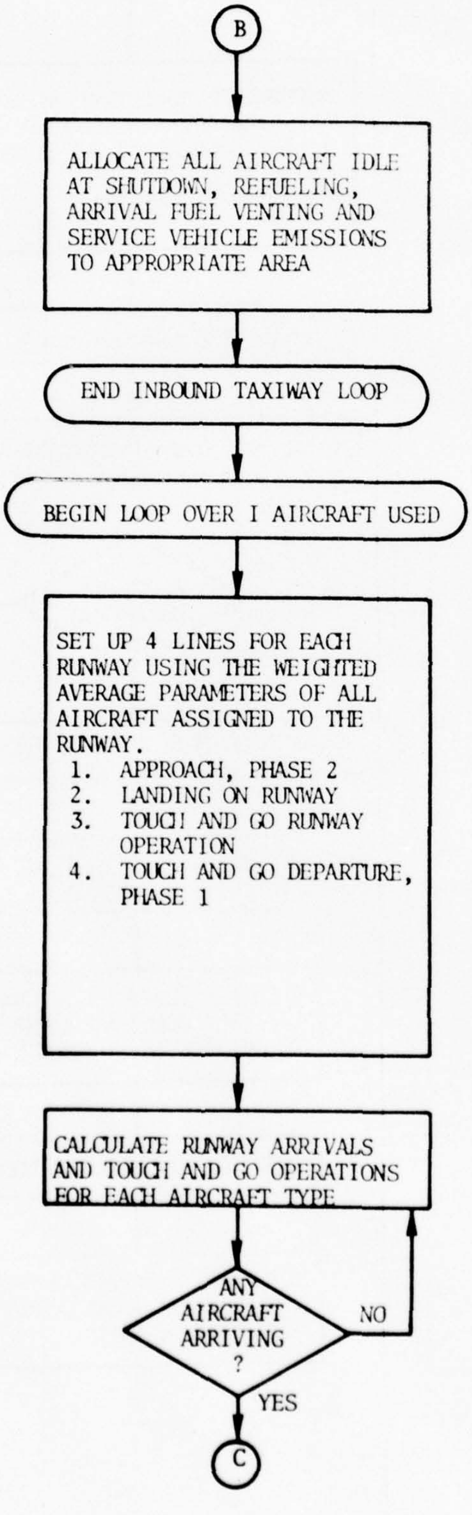

START
IFLAG
= 0
E
≠ 0
READ ARRFCN AND DEPFCN FROM TAPE
FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY
CALL FLTPTH TO DETERMINE WEIGHTED AVERAGE VELOCITIES, ANGLES AND ALTITUDES OF ALL AIRCRAFT TYPES USING EACH RUNWAY
SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS
BEGIN LOOP OVER N RUNWAYS
IS RUNWAY USED ?
NO
YES
A

SUBROUTINE ACSRCE (Cont'd.)
(APPLICATIONS MODEL)


A
SET TOUCHDOWN POINT FOR TOUCH AND GO OPERATIONS TO THAT USED FOR LANDING ON RUNWAY SO THAT THE SAME APPROACH PATH CAN BE USED AS FOR ARRIVALS.
BEGIN LOOP OVER J INBOUND TAXIWAYS
ANY AIRCRAFT ARRIVING ?
NO
YES
BEGIN LOOP OVER K TAXIWAY SEGMENT
SET UP SEGMENT LINE SOURCE GEOMETRIES
ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS TO APPROPRIATE SEGMENTS
END TAXIWAY SEGMENT LOOP
DETERMINE AIRCRAFT INBOUND PARKING AREA ASSOCIATED WITH TAXIWAY PATH
B

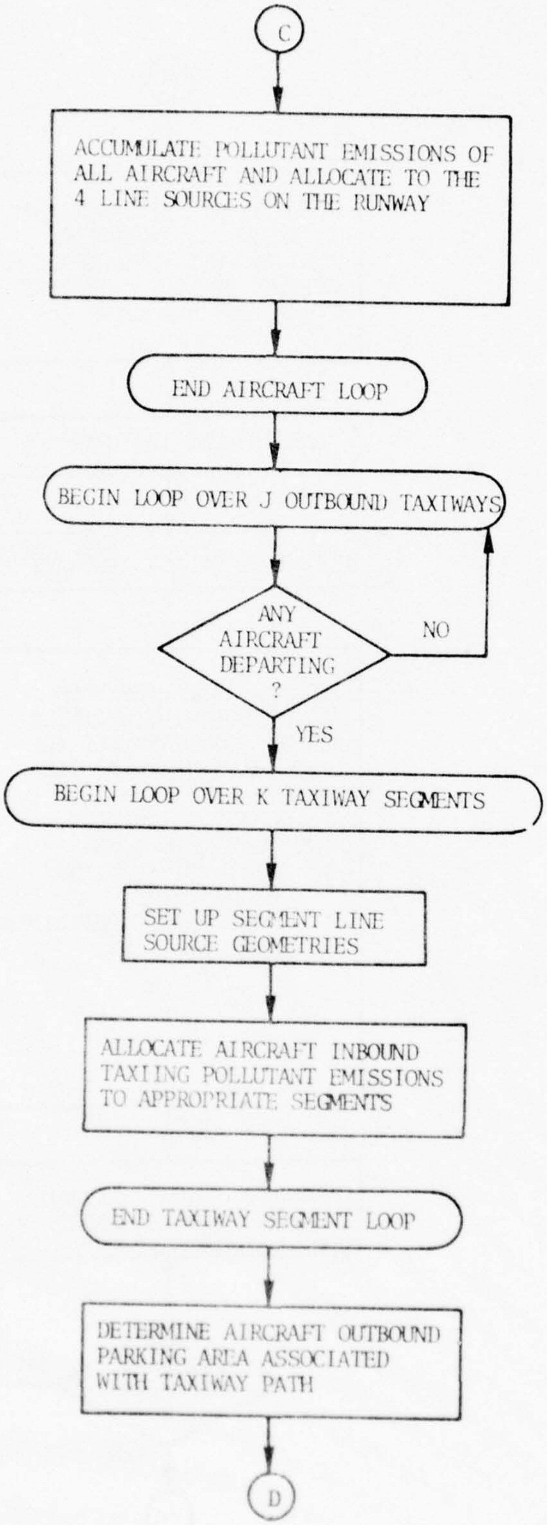
SUBROUTINE ACSRCE (Cont'd.)

(APPLICATIONS MODEL)

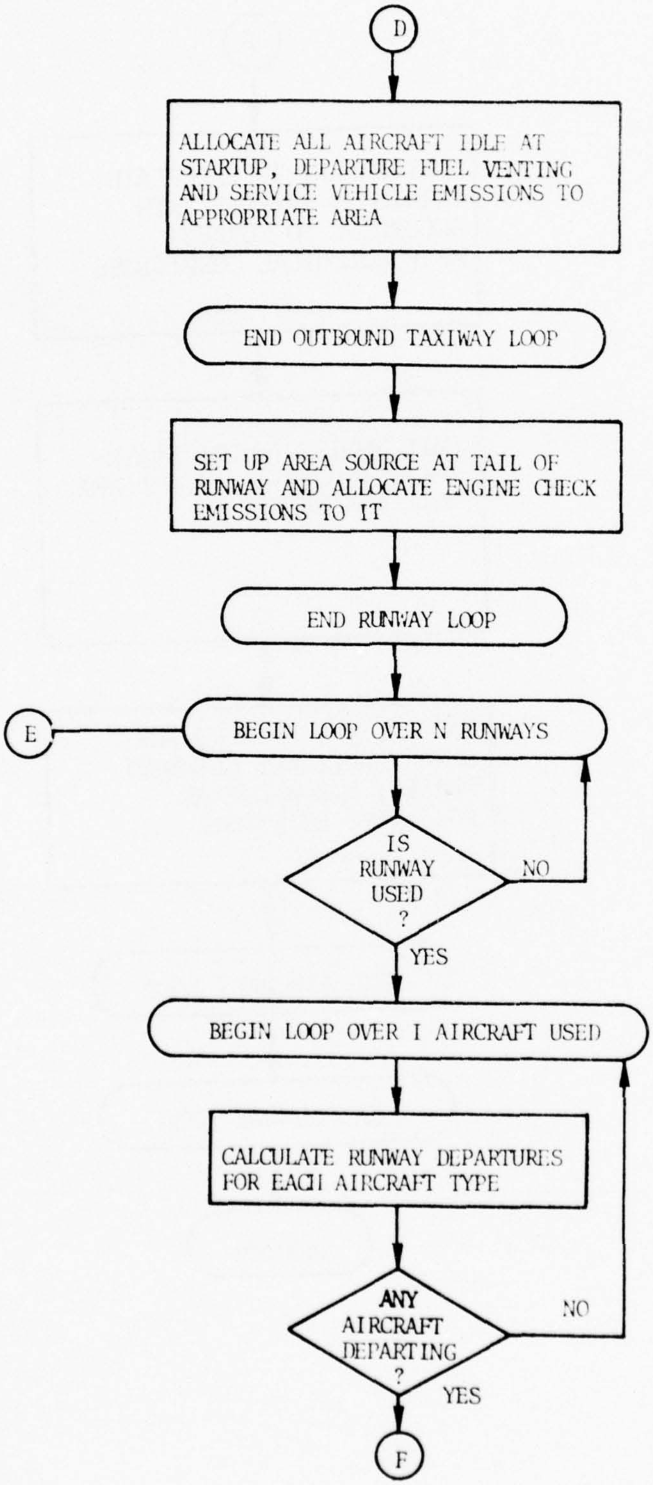

B
ALLOCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING, ARRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
END INBOUND TAXIWAY LOOP
BEGIN LOOP OVER I AIRCRAFT USED
SET UP 4 LINES FOR EACH RUNWAY USING THE WEIGHTED AVERAGE PARAMETERS OF ALL AIRCRAFT ASSIGNED TO THE RUNWAY.
1. APPROACH, PHASE 2
2. LANDING ON RUNWAY
3. TOUCH AND GO RUNWAY OPERATION
4. TOUCH AND GO DEPARTURE, PHASE 1
CALCULATE RUNWAY ARRIVALS AND TOUCH AND GO OPERATIONS FOR EACH AIRCRAFT TYPE
ANY AIRCRAFT ARRIVING ?
NO
YES
C

SUBROUTINE ACSRCE (Cont'd.)

(APPLICATIONS MODEL)

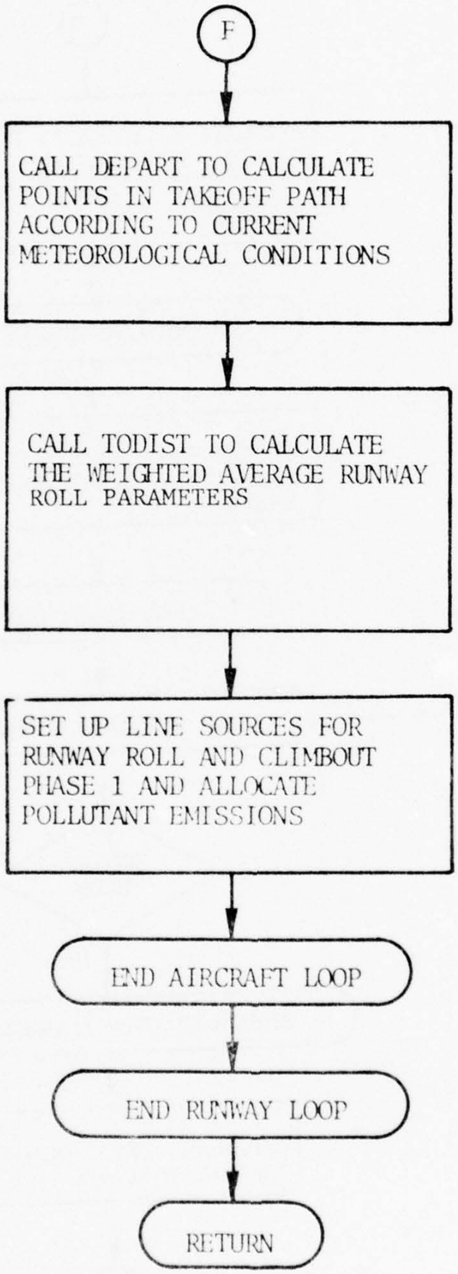

C
ACCUMULATE POLLUTANT EMISSIONS OF ALL AIRCRAFT AND ALLOCATE TO THE 4 LINE SOURCES ON THE RUNWAY
END AIRCRAFT LOOP
BEGIN LOOP OVER J OUTBOUND TAXIWAYS
ANY AIRCRAFT DEPARTING ?
NO
YES
BEGIN LOOP OVER K TAXIWAY SEGMENTS
SET UP SEGMENT LINE SOURCE GEOMETRIES
ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS TO APPROPRIATE SEGMENTS
END TAXIWAY SEGMENT LOOP
DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED WITH TAXIWAY PATH
D

SUBROUTINE ACSRCE (Cont'd.)
(APPLICATIONS MODEL)


D
ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA
END OUTBOUND TAXIWAY LOOP
SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE ENGINE CHECK EMISSIONS TO IT
END RUNWAY LOOP
E
BEGIN LOOP OVER N RUNWAYS
IS RUNWAY USED ?
NO
YES
BEGIN LOOP OVER I AIRCRAFT USED
CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE
ANY AIRCRAFT DEPARTING ?
NO
YES
F

SUBROUTINE ACSRCE (Cont'd.)

(APPLICATIONS MODEL)


F
CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING TO CURRENT METEOROLOGICAL CONDITIONS
CALL TODIST TO CALCULATE THE WEIGHTED AVERAGE RUNWAY ROLL PARAMETERS
SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT PHASE 1 AND ALLOCATE POLLUTANT EMISSIONS
END AIRCRAFT LOOP
END RUNWAY LOOP
RETURN

```
C      PROGRAM ACSRCE
       SUBROUTINE ACSRCE
C
C      THIS ROUTINE SETS UP THE AIRCRAFT SOURCE ARRAYS
C      AND ALLOCATES THE POLLUTANT EMISSIONS TO THE
C      APPROPRIATE AREA OR LINE
C
C    THIS VERSION OF ACSRCE HAS BEEN MODIFIED FOR USE
C    IN THE APPLICATIONS MODEL OF THE LONG TERM
C    EMISSION/DISPERSION CODE
C
C
       REAL LNDSPD
       INTEGER  ENGNO
       COMMON /RECPT/ MRECPT,MAXFIL
       COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,
     . NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),
     . ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)
       COMMON /ACEDP1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),
     . LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),
     . COSPD2(8),STTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)
       COMMON /ACEDP2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),
     . ACSTIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),
     . IIBSEG(16,8,6),IDIBTW(8,6),ITARFR(8,8,6),NOBTT(6),NOBSEG(8,6),
     . IOBSEG(16,8,6),IOBTW(8,6),TTDPFR(8,8,6),NRASQ(6),IDPRKA(6),
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)
       COMMON / MET / WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,
     1 TEMK
       COMMON / DEFALT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)
       COMMON /DSTRPT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),
     . VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),
     . CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1
       COMMON /PERIOD/ IMO    ,NODAYS,IDY ,IHR1,IHR2,IFLAG,JFLAG
       COMMON /MODSTM/ NPRL,NCOP,TST,DISA,RDP(6),RAR(6),RTG(6),
     . HTAPP(6),HITGO(6),VELLND(6),VELTXI(6),ANGAPP(6),ANGTGO(6),
     . DIST(2,6),TOPT(2,6),RRTO(2,6),RRTXI(2,6),HTCO(2,6),ANGCO(2,6)
     . ,FRAC(8)
       DIMENSION IACAR(2,18),          PARFCT(18),APARSQ(6,3),NQ(25)
       XP(XC,YC,W)=YC*SIN(W)+XO
       YP(YC,YC,W)=YC*COS(W)+YO
       DAYS=NODAYS
       NT=NELTS+5
       IWIND=29+IWD
C
C      AN IFLAG OF 0 MEANS THAT ALL AIRCRAFT SOURCES EXCEPT
C      FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 REMAIN
C      UNCHANGED
C
       IF(IFLAG.EQ.0) GO TO 69
C
C      READ ARRFCN AND DEPFCN FROM TAPE
C
       IF (IWD.GE.1.AND.IWD.LE.MAXFIL) GO TO 1000
       PRINT 9000,MRECPT,MAXFIL,IWD
 9000  FORMAT(29H0FILE REQUEST ERROR IN ACSRCE,3I5)
       GO TO 1040
 1000  IF (MRECPT-IWD) 1010,1030,1020
 1010  READ (30)
       MRECPT=MRECPT+1
```

```
      GO TO 1000                                                        ACSRC062
 1020 REWIND 30                                                         ACSRC063
      MRECPT=1                                                          ACSRC064
      GO TO 1000                                                        ACSRC065
 1030 READ (30) ARRFCN,DEPFCN                                           ACSRC066
      MRECPT=MRECPT+1                                                   ACSRC067
 1040 CONTINUE                                                          ACSRC068
C                                                                       ACSRC069
C     FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL                ACSRC070
C     DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY                         ACSRC071
C                                                                       ACSRC072
      NHI=IHR2                                                          ACSRC073
      IF(IHR1.GT.IHR2) NHI=24+IHR2                                      ACSRC074
      HRS=NHI-IHR1+1                                                    ACSRC075
      DO 5 I=1,NACTYP                                                   ACSRC076
      HRFRC=0.                                                          ACSRC077
      DO 4 JJ=IHR1,NHI                                                  ACSRC078
      J=JJ                                                              ACSRC079
      IF(JJ.GT.24) J=JJ-24                                              ACSRC080
    4 HRFRC=HRFRC+ACHR(J,I)                                             ACSRC081
      HRFRC=HRFRC/HRS                                                   ACSRC082
      FRAC(I)=ACMO(IMO,I)*ACDY(IDY,I)*HRFRC*7.0/DAYS*(1.E+6/3.6)        ACSRC083
    5 CONTINUE                                                          ACSRC084
C                                                                       ACSRC085
C   CALL FLIPTH TO DETERMINE WEIGHTED AVERAGE VELOCITIES, ANGLES        ACSRC086
C   AND ALTITUDES OF ALL AIRCRAFT TYPES USING EACH RUNWAY               ACSRC087
C                                                                       ACSRC088
      CALL FLIPTH                                                       ACSRC089
    8 NACPT=0                                                           ACSRC090
      NB=0                                                              ACSRC091
      NC=0                                                              ACSRC092
      NZ=0                                                              ACSRC093
C                                                                       ACSRC094
C     SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS          ACSRC095
C                                                                       ACSRC096
      DO 1  L=1,NPKAR                                                   ACSRC097
      NSQ=NPASQ(L)                                                      ACSRC098
      SPARSQ=0.0                                                        ACSRC099
      DO 2 J=1,NSQ                                                      ACSRC100
      NB=NB+1                                                           ACSRC101
      ACAR(1,NB)=PAREA(L,J,1)                                           ACSRC102
      ACAR(2,NB)=PAREA(L,J,2)                                           ACSRC103
      ACAR(3,NB)=ACLNDZ/2.                                              ACSRC104
      ACAR(4,NB)=PAREA(L,J,3) *1000.                                    ACSRC105
      APARSQ(L,J)=ACAR(4,NB) ** 2                                       ACSRC106
      SPARSQ = SPARSQ + APARSQ(L,J)                                     ACSRC107
      ACAR(5,NB)=ACLNDZ                                                 ACSRC108
      IACAR(1,NB)=IDPRKA(L)                                             ACSRC109
    2 IACAR(2,NB)=NSQ                                                   ACSRC110
      DO 91 J=1,NSQ                                                     ACSRC111
      NZ=NZ+1                                                           ACSRC112
   91 PARFCT(NZ) = APARSQ(L,J) / SPARSQ                                 ACSRC113
    1 CONTINUE                                                          ACSRC114
C                                                                       ACSRC115
      DO 93 I=1,NLSEGS                                                  ACSRC116
   93 NQ(I)=0                                                           ACSRC117
      NPKSRC=NB                                                         ACSRC118
      DO 3  L=1,NPKSRC                                                  ACSRC119
      DO 3  K=6,NT                                                      ACSRC120
    3 ACAR(K,L)=0.0                                                     ACSRC121
      TVP= EXP(ALPHA(2)-BETA(2)/TEMK)                                   ACSRC122
C                                                                       ACSRC123
```

```
C       BEGIN LOOP OVER N RUNWAYS
C
        DO 10 N=1,NRNWYS
C
C       IS RUNWAY USED WITH THIS WIND DIRECTION?
C
        IF (IUSWD(IWD,N).EQ.0) GO TO 10
C
C     SET TOUCHDOWN POINT FOR TOUCH AND GO OPERATIONS
C     TO THAT USED FOR LANDING ON RUNWAY SO THAT THE
C     SAME APPROACH PATH CAN BE USED AS FOR ARRIVALS
C
        THETA=RNWY(7,N)
        XC=0.3048*SIN(THETA)+RNWY(2,N)
        YC=0.3048*COS(THETA)+RNWY(3,N)
        NTT=NIBTI(N)
        IF(NTT.EQ.0) GO TO 50
C
C       BEGIN LOOP OVER J INBOUND TAXIWAYS
C
        DO 11 J=1,NTT
C
C       ANY AIRCRAFT ARRIVING ON THIS RUNWAY?
C
        DO 7 I=1,NACTYP
        IF(TTARFF(J,I,N)*ARRFCN(23,I,N).GT.0.0) GO TO 701
      7 CONTINUE
        GO TO 11
    701 NSGLNS = NIBSEG (J,N)
C
C       BEGIN LOOP OVER K TAXIWAY SEGMENTS
C
        DO 12 K=1,NSGLNS
C
C       SET UP SEGMENT LINE SOURCE GEOMETRIES
C
        JJ = IIBSEG(K,J,N)
        IF(NC(JJ).NE.0) GO TO 130
        NC=NC+1
        NC(JJ)=NC
        DO 121 L=1,12
    121 ACLN(L,NC)=ACLNSG(L,JJ)
        ACLN(9,NC)=1.0
        ACLN(10,NC)=1.0
C
C       ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS
C       TO APPROPRIATE SEGMENTS
C
        DO 13 L=1,NPLTS
        LL=L+12
   13   ACLN(LL,NC)=0.0
    130 ND=NC(JJ)
        DO 14 I=1,NACTYP
        AA=ENGNO(I ,1)
        IF(IFGFLG.GT.0) AA=ENGNO(I ,2)
        ARR=TTARFF(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
        IF(ARR.LE.0.0)GO TO 14
        TIME=ACLN(11,ND)/(TXISPD(I )*ACLNSG(9,JJ))
        FRC=AA*ARR*TIME*FRAC(I )
        DO 15 L=1,NPLTS
        KK=L+12
   15   ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I ,2,L)
```

```
 14   CONTINUE
  12  CONTINUE
C
C      END TAXIWAY SEGMENT LOOP
C
C
C      DETERMINE AIRCRAFT INBOUND PARKING AREA
C      ASSOCIATED WITH TAXIWAY PATH
C
       DO 16 I=1,NPKSRC
       II=I
       IDPK=IACAR(1,I)
       IF(IDPK.EQ.IDIBPA(J,N))GO TO 17
  16  CONTINUE
       PRINT 18, IDIBPA(J,N),J,N
 18   FORMAT ('0INBOUND PARKING AREA 'I3,'OF TAXIWAY='I3,'; RUNWAY='I3,'
     1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')
       STOP
 17   CONTINUE
C
C      ALLOCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING,
C      ARRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS
C      TO APPROPRIATE AREA
C
       NSQ=IACAR(2,II)
       DO  19 I=1,NACTYP
       ARR=TTARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)
       IF(ARR.LE.0.0) GO TO 19
       AA=ENGNO(I,1)
       IF(IEGFLG.GT.0) AA=ENGNO(I,2)
       TIME=SHTDNT(I)/60.
       FRC=AA*ARR*TIME*FRAC(I)
       TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)
       DO 20 L=1,NSQ
       JJ=II+L-1
       DO 21 K=1,NPLTS
       KK=K+5
       ACAR(KK,JJ)=ACAR(KK,JJ)+FRC*ACEMFC(I,1,K) * PARFCT(JJ)
       ACAR(KK,JJ)=ACAR(KK,JJ) + (ARSVEM(K,I,1) + ARSVEM(K,I,2) +
     .ARSVEM(K,I,3) + ARSVEM(K,I,4)+ARSVEM(K,I,5)) * ARR * FRAC(I)
     . * PARFCT(JJ)
       IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+(0.3*TVP*ACFUEL(I)*0.5
     1/1000. + ACSPIL(I) + ARFLVT(I)) * ARR * FLDENS(JES1(I)) * FRAC(I)
     . * PARFCT(JJ)
 21   CONTINUE
 20   CONTINUE
 19   CONTINUE
 11   CONTINUE
C
C      END INBOUND TAXIWAY LOOP
C
C
C      BEGIN LOOP OVER I AIRCRAFT USED
C
C
C    SET UP 4 LINES FOR EACH RUNWAY USING THE WEIGHTED AVERAGE
C    PARAMETERS OF ALL AIRCRAFT ASSIGNED TO THE RUNWAY
C
       DO 25 J=1,4
       JJ=J+NC
       DO 26 K=1,18
   26 ACLN(K,JJ)=0.0
```

```
      ACLN(3,JJ)=ACLNDZ/2.
      ACLN(4,JJ)=ACLNDY
      ACLN(5,JJ)=ACLNDZ
      ACLN(8,JJ)=ACLNDZ/2.
      ACLN(9,JJ)=1.0
      ACLN(10,JJ)=1.0
   25 ACLN(12,JJ)=1.0
C
C    LINE 1 - APPROACH, PHASE 2
C
      YAP=-HTAPP(N)/TAN(ANGAPP(N))
      ACLN(1,NC+1)=XP(XO,YAP,THETA)
      ACLN(2,NC+1)=YP(YO,YAP,THETA)
      ACLN(3,NC+1)=HTAPP(N)*1000.0
      ACLN(6,NC+1)=XO
      ACLN(7,NC+1)=YO
      ACLN(11,NC+1)=HTAPP(N)/SIN(ANGAPP(N))
      ACLN(12,NC+1)=ACLN(11,NC+1)/ACLN(9,NC+1)
C
C    LINE 2 - LANDING ON RUNWAY
C
      YLN=DISRNW(N)-0.3048
      ACLN(1,NC+2)=XO
      ACLN(2,NC+2)=YO
      ACLN(6,NC+2)=XP(XO,YLN,THETA)
      ACLN(7,NC+2)=YP(YO,YLN,THETA)
      ACLN(9,NC+2)=VELLND(N)
      ACLN(10,NC+2)=VELTXI(N)
      ACLN(11,NC+2)=YLN
      ACLN(12,NC+2)=2.*YLN/(VELLND(N)+VELTXI(N))
C
C    LINE 3 - TOUCH AND GO RUNWAY OPERATION
C
      ACLN(1,NC+3)=XO
      ACLN(2,NC+3)=YO
      ACLN(6,NC+3)=XP(XO,0.3048,THETA)
      ACLN(7,NC+3)=YP(YO,0.3048,THETA)
      ACLN(11,NC+3)=0.3048
      ACLN(12,NC+3)=0.3048/ACLN(9,NC+3)
C
C    LINE 4 - TOUCH AND GO DEPARTURE, PHASE 1
C
      ACLN(1,NC+4)=ACLN(6,NC+3)
      ACLN(2,NC+4)=ACLN(7,NC+3)
      YCOR=0.3048+HTTGO(N)/TAN(ANGTGO(N))
      ACLN(6,NC+4)=XP(XO,YCOR,THETA)
      ACLN(7,NC+4)=YP(YO,YCOR,THETA)
      ACLN(8,NC+4)=HTTGO(N)*1000.0
      ACLN(11,NC+4)=HTTGO(N)/SIN(ANGTGO(N))
      ACLN(12,NC+4)=ACLN(11,NC+4)/ACLN(9,NC+4)
C
C    CALCULATE RUNWAY ARRIVALS AND TOUCH AND GO
C    OPERATIONS FOR EACH AIRCRAFT TYPE
C
      DO 30 I=1,NACTYP
      AA=ENGNO(I,1)
      ARR=ARRFCN(22,I,N)*ANNARR(I)
      ATG=ARRFCN(23,I,N)*ANNTGO(I)
      ATOT=ARR+ATG
      IF (ATOT.LE.0.0) GO TO 30
C
C    ACCUMULATE POLLUTANT EMISSIONS OF ALL AIRCRAFT AND
```

```
C     ALLOCATE TO THE 4 LINE SOURCES ON THE RUNWAY                         ACSRC310
C                                                                          ACSRC311
      DO 31 J=1,4                                                          ACSRC312
      JJ=J+NC                                                              ACSRC313
      GO TO (32,33,37,34),J                                                ACSRC314
   32 JMODE=8                                                              ACSRC315
      ACT=ATOT*ARRFCN(12,I,N)*FRAC(I)*AA                                   ACSRC316
      TIME=ARRFCN(12,I,N)                                                  ACSRC317
      GO TO 35                                                             ACSRC318
   33 IF (ARR.LE.0.0) GO TO 31                                             ACSRC319
      JMODE=9                                                              ACSRC320
      ACT=ARR*ARRFCN(18,I,N)*FRAC(I)*AA                                    ACSRC321
      GO TO 35                                                             ACSRC322
   34 IF (ATG.LE.0.0) GO TO 31                                             ACSRC323
      JMODE=5                                                              ACSRC324
      TGOTIM=2.*(YCOR-.3048)/(COSPD1(I)+TOSPD(I))                          ACSRC325
      ACT=ATG*TGOTIM*FRAC(I)*AA                                            ACSRC326
   35 DO 36 K=1,NELTS                                                      ACSRC327
      KK=K+12                                                              ACSRC328
   36 ACLN(KK,JJ)=ACLN(KK,JJ)+ACT*ACEMFC(I,JMODE,K)                        ACSRC329
      GO TO 31                                                             ACSRC330
   37 IF (ATG.LE.0.0) GO TO 31                                             ACSRC331
      TIME=2.*ACLN(11,JJ)/(1.3*LNDSPD(I)+0.7*TOSPD(I))                     ACSRC332
      ACT=ATG*TIME*FRAC(I)*AA                                              ACSRC333
      DO 38 K=1,NELTS                                                      ACSRC334
      KK=K+12                                                              ACSRC335
   38 ACLN(KK,JJ)=ACLN(KK,JJ)+ACT*(0.3*ACEMFC(I,9,K)+0.7*                  ACSRC336
     . ACEMFC(I,4,K))                                                      ACSRC337
   31 CONTINUE                                                             ACSRC338
   30 CONTINUE                                                             ACSRC339
      NC=NC+4                                                              ACSRC340
C                                                                          ACSRC341
C     END AIRCRAFT LOOP                                                    ACSRC342
C                                                                          ACSRC343
   50 NTT=NOBIT(N)                                                         ACSRC344
      IF(NTT.EQ.0) GO TO 10                                                ACSRC345
C                                                                          ACSRC346
C     BEGIN LOOP OVER J OUTBOUND TAXIWAYS                                  ACSRC347
C                                                                          ACSRC348
      DO 51 J=1,NTT                                                        ACSRC349
C                                                                          ACSRC350
C     ANY AIRCRAFT DEPARTING ON THIS TAXIWAY?                              ACSRC351
C                                                                          ACSRC352
      DO 6 I=1,NACTYP                                                      ACSRC353
      IF(TIDEPR(J,I,N)*DEPFCN(23,I,N).GT.0.0) GO TO 601                    ACSRC354
    6 CONTINUE                                                             ACSRC355
      GO TO 51                                                             ACSRC356
  601 NSGLNS=NOBSEG(J,N)                                                   ACSRC357
C                                                                          ACSRC358
C     BEGIN LOOP OVER K TAXIWAY SEGMENTS                                   ACSRC359
C                                                                          ACSRC360
      DO 52 K=1,NSGLNS                                                     ACSRC361
C                                                                          ACSRC362
C     SET UP SEGMENT LINE SOURCE GEOMETRIES                                ACSRC363
C                                                                          ACSRC364
      JJ=IOBSEG(K,J,N)                                                     ACSRC365
      IF(NQ(JJ).NE.0) GO TO 131                                            ACSRC366
      NC=NC+1                                                              ACSRC367
      NQ(JJ)=NC                                                            ACSRC368
      DO 122 L=1,12                                                        ACSRC369
  122 ACLN(L,NC)=ACLNSG(L,JJ)                                              ACSRC370
      ACLN(9,NC)=1.0                                                       ACSRC371
      ACLN(10,NC) =1.0                                                  ACSRC372
C                                                                       ACSRC373
C     ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS             ACSRC374
C     TO APPROPRIATE SEGMENTS                                           ACSRC375
C                                                                       ACSRC376
      DO 53 L=1,NPLTS                                                   ACSRC377
      LL=L+12                                                           ACSRC378
   53 ACLN(LL,NC) =0.0                                                  ACSRC379
  131 ND=NC(JJ)                                                         ACSRC380
      DO 54 I=1,NACTYP                                                  ACSRC381
      DEP=TIDPFR(J,I,N) *DEPFCN(23,I,N) *ANNDEP(I)                      ACSRC382
      IF(DEP.LE.0.0) GO TO 54                                           ACSRC383
      AA=ENGNO(I,1)                                                     ACSRC384
      IF(IEGFLG.GT.0)  AA=ENGNO(I,2)                                    ACSRC385
      TIME= ACLN(11,ND) / TXISPD(I)                                     ACSRC386
      FRC= AA* DEP*TIME*FRAC(I)                                         ACSRC387
      DO 55 L=1,NPLTS                                                   ACSRC388
      KK=L+12                                                           ACSRC389
   55 ACLN(KK,ND)= ACLN(KK,ND)+FRC*ACEMFC(I,2,L)                        ACSRC390
   54 CONTINUE                                                          ACSRC391
   52 CONTINUE                                                          ACSRC392
C                                                                       ACSRC393
C     END TAXIWAY SEGMENT LOOP                                          ACSRC394
C                                                                       ACSRC395
C                                                                       ACSRC396
C     DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED               ACSRC397
C     WITH TAXIWAY PATH                                                 ACSRC398
C                                                                       ACSRC399
      DO 56 I=1,NPKSRC                                                  ACSRC400
      II =I                                                             ACSRC401
      IDPK=IACAR(1,I)                                                   ACSRC402
      IF(IDPK.EQ.IDOBPA(J,N)) GO TO 58                                  ACSRC403
   56 CONTINUE                                                          ACSRC404
      PRINT 57,IDOBPA(J,N),J,N                                          ACSRC405
   57 FORMAT(22H0OUTBOUND PARKING AREA,I3,11H OF TAXIWAY,I3,8H, RUNWAY, ACSRC406
     . I3,47H IS NOT CONSISTENT WITH PARKING AREA ID NUMBERS)           ACSRC407
      STOP                                                              ACSRC408
C                                                                       ACSRC409
C     ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL             ACSRC410
C     VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA         ACSRC411
C                                                                       ACSRC412
   58 NSQ=IACAR(2,II )                                                  ACSRC413
      DO 59 I=1,NACTYP                                                  ACSRC414
      DEP=TIDPFR(J,I,N) *DEPFCN(23,I,N) *ANNDEP(I)                      ACSRC415
      IF(DEP.EQ.0.0) GO TO 59                                           ACSRC416
      AA=ENGNO(I,1)                                                     ACSRC417
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)                                     ACSRC418
      TIME=SRTUPT(I)/60.                                                ACSRC419
      FRC = AA* DEP* TIME * FRAC(I)                                     ACSRC420
      TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)                    ACSRC421
      DO 60 L=1,NSQ                                                     ACSRC422
      JJ =II +L-1                                                       ACSRC423
      DO 61 K=1,NPLTS                                                   ACSRC424
      KK=K+5                                                            ACSRC425
      ACAR(KK,JJ) = ACAR(KK,JJ) + ((FRC * ACEMFC(I,1,K)) +              ACSRC426
     . ((DPSVEM(K,I,1) + DPSVEM(K,I,2) + DPSVEM(K,I,3) + DPSVEM(K,I,4)  ACSRC427
     . + DPSVEM(K,I,5)) * DEP * FRAC(I))) * PARFCT(JJ)                  ACSRC428
      IF (K.EQ.2) ACAR(KK,JJ) = ACAR(KK,JJ) + DPFLVT(I) * DEP * FLDENS( ACSRC429
     . JES1(I)) * FRAC(I) * PARFCT(JJ)                                  ACSRC430
   61 CONTINUE                                                          ACSRC431
   60 CONTINUE                                                          ACSRC432
   59 CONTINUE                                                          ACSRC433
   51 CONTINUE
C
C     END OUTBOUND TAXIWAY LOOP
C
      NB=NB+1
C
C     SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE
C     ENGINE CHECK EMISSIONS TO IT
C
      ACAR(1,NB)=RNWY(2,N)-.05 * SIN(THETA)
      ACAR(2,NB)=RNWY(3,N)-.05*COS(THETA)
      ACAR(3,NB)= ACLNDZ/2.
      ACAR(4,NB)= 100.0
      ACAR(5,NB)= ACLNDZ
      DO 65 K=1,NPLTS
      KK=K+5
   65 ACAR(KK,NB)=0.0
      DO 66 I=1,NACTYP
      DEP=DEPFCN(23,I,N)*ANNDEP(I)
      IF(DEP.EQ.0.0) GO TO 66
      AA=ENGNO(I,1)
      IF(IEGFLG.GT.0) AA=ENGNO(I,2)
      TIME= EGCHKT(I)/60.
      FRC= TIME *DEP*AA*FRAC(I)
      DO 67 K=1,NPLTS
      KK=K+5
   67 ACAR(KK,NB)=ACAR(KK,NB)+ FRC* ACEMFC(I,3,K)
   66 CONTINUE
   10 CONTINUE
C
C     END RUNWAY LOOP
C
      NACAR=NB
      NC1=NC
   69 NC=NC1
C
C     BEGIN LOOP OVER N RUNWAYS
C
      DO 79 N=1,NRNWYS
C
C     IS RUNWAY USED WITH THIS WIND DIRECTION?
C
      IF(IUSWD(IWD,N).EQ.0) GO TO 79
      XC=RNWY(2,N)
      YC=RNWY(3,N)
      THETA=RNWY(7,N)
C
C     BEGIN LOOP OVER I AIRCRAFT USED
C
      DO 70 I=1,NACTYP
C
C     CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE
C
      DEP=DEPFCN(23,I,N)*ANNDEP(I)
C
C     ANY AIRCRAFT DEPARTING FROM THIS RUNWAY?
C
      IF(DEP.EQ.0.0) GO TO 70
C
C     CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING
C     TO CURRENT METEOROLOGICAL CONDITIONS
C
```

```
      CALL DEPART (N,I)
   70 CONTINUE
C
C  CALL TCDIST TO CALCULATE THE WEIGHTED AVERAGE
C  RUNWAY ROLL PARAMETERS
C
      CALL TCDIST(N)
C
C  SET UP THE CLIMBOUT AND RUNWAY ROLL LINES FOR EACH
C  RUNWAY USING THE WEIGHTED AVERAGE PARAMETERS FOR
C  ALL AIRCRAFT ASSIGNED TO THE RUNWAY
C
      NHI=NCOP+NRRL
      DO 75 J=1,NHI
      JJ=J+NC
      DO 76 K=1,18
   76 ACLN(K,JJ)=0.0
      ACLN(3,JJ)=ACLNDZ/2.
      ACLN(4,JJ)=ACLNDY
      ACLN(5,JJ)=ACLNDZ
      ACLN(8,JJ)=ACLNDZ/2.
      ACLN(9,JJ)=1.0
   75 ACLN(10,JJ)=1.0
C
C  CLIMBOUT, PHASE 1 - A MAXIMUM OF 2 LINES ARE CREATED
C
      DO 77 J=1,NCOP
      JJ=NC+J
      ACLN(1,JJ)=XP(XO,TOPT(J,N),THETA)
      ACLN(2,JJ)=YP(YO,TOPT(J,N),THETA)
      YCOP=TOPT(J,N)+HTCO(J,N)/TAN(ANGCO(J,N))
      ACLN(6,JJ)=XP(XO,YCOP,THETA)
      ACLN(7,JJ)=YP(YO,YCOP,THETA)
      ACLN(8,JJ)=  HTCO(J,N)*1000.0
      ACLN(11,JJ)=   HTCO(J,N)/SIN(ANGCO(J,N))
   77 ACLN(12,JJ)=   ACLN(11,JJ)/ACLN(9,JJ)
C
C  RUNWAY ROLL - A MAXIMUM OF 2 LINES ARE CREATED
C
      DO 78 J=1,NRRL
      JJ=NC+NCOP+J
      ACLN(1,JJ)=XO
      ACLN(2,JJ)=YO
      ACLN(6,JJ)=   XP(XO,DIST(J,N),THETA)
      ACLN(7,JJ)=   YP(YO,DIST(J,N),THETA)
      ACLN(9,JJ)=   RRTXI(J,N)
      ACLN(10,JJ)=   RRTO(J,N)
      ACLN(11,JJ)=   DIST(J,N)
   78 ACLN(12,JJ)=   2.*DIST(J,N)/(RRTXI(J,N)+RRTO(J,N))
C
C  ACCUMULATE POLLUTANT EMISSIONS OF ALL AIRCRAFT AND
C  ALLOCATE TO THE CLIMBOUT AND RUNWAY ROLL LINE SOURCES
C
      DO 80 I=1,NACTYP
      DEP=DEPFCN(23,I,N)*ANNDEP(I)
      IF (DEP.LE.0.0) GO TO 80
      AA=ENGNO(I,1)
      DO 81 L=1,2
      IF (L.EQ.2) GO TO 82
      JMODE=5
      ACT=AA*DEP*DEPFCN(12,I,N)*FRAC(I)
      J=1
      IF (ASCNT1(I).GT.TST) J=2                                         ACSRC558
      JJ=NC+J                                                           ACSRC559
      GO TO 83                                                          ACSRC560
   82 JMODE=4                                                           ACSRC561
      ACT=AA*DEP*DEPFCN(6,I,N)*FRAC(I)                                  ACSRC562
      J=1                                                               ACSRC563
      IF (DEPFCN(5,I,N).GT.DISA) J=2                                    ACSRC564
      JJ=NC+NCOP+J                                                      ACSRC565
   83 DO 85 K=1,NPLTS                                                   ACSRC566
      KK=K+12                                                           ACSRC567
   85 ACLN(KK,JJ)=ACLN(KK,JJ)+ACT*ACEMFC(I,JMODE,K)                     ACSRC568
   81 CONTINUE                                                          ACSRC569
   80 CONTINUE                                                          ACSRC570
      NC=NC+NCOP+NRRL                                                   ACSRC571
  79  CONTINUE                                                          ACSRC572
C                                                                       ACSRC573
C     END RUNWAY LOOP                                                   ACSRC574
C                                                                       ACSRC575
      NACLN=NC                                                          ACSRC576
      RETURN                                                            ACSRC577
      END                                                               ACSRC578
```

SUBROUTINE FLTPTH

Purpose:

To define the flight path parameters for all runways.

Input:

Aircraft activity data on all runways.

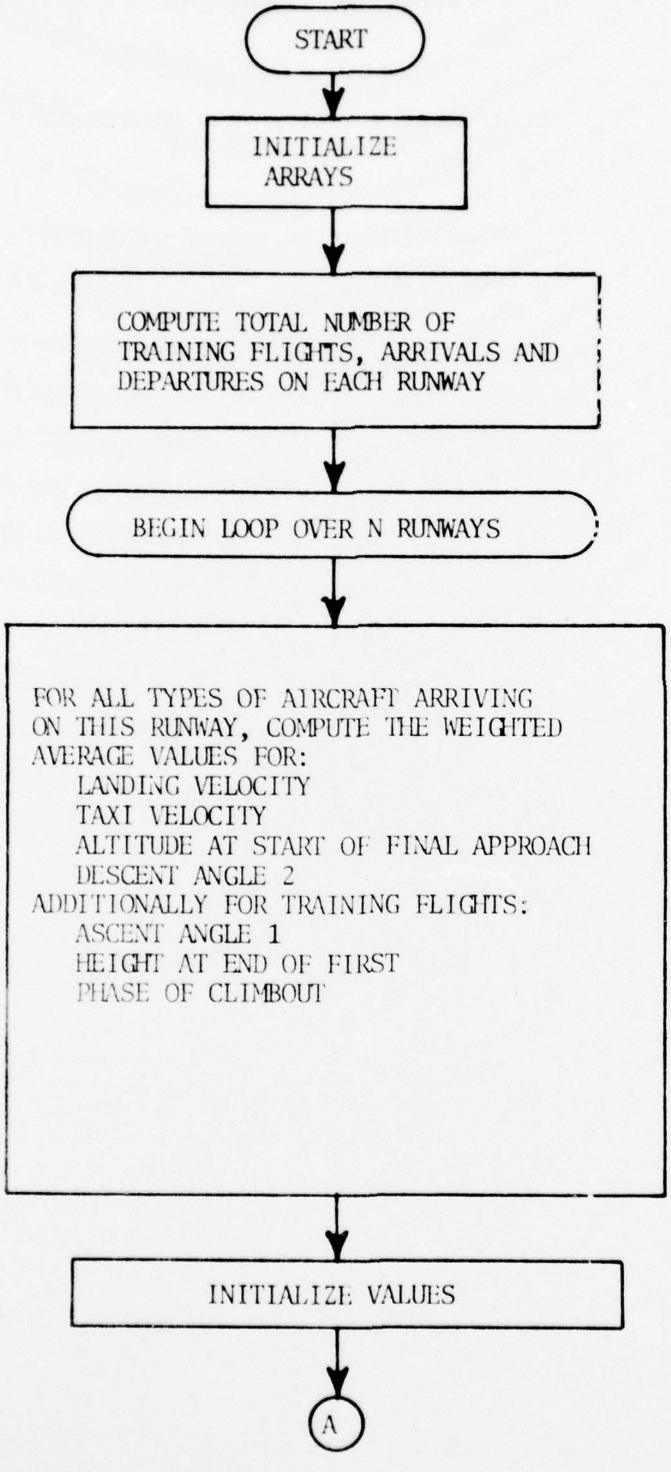
Output:

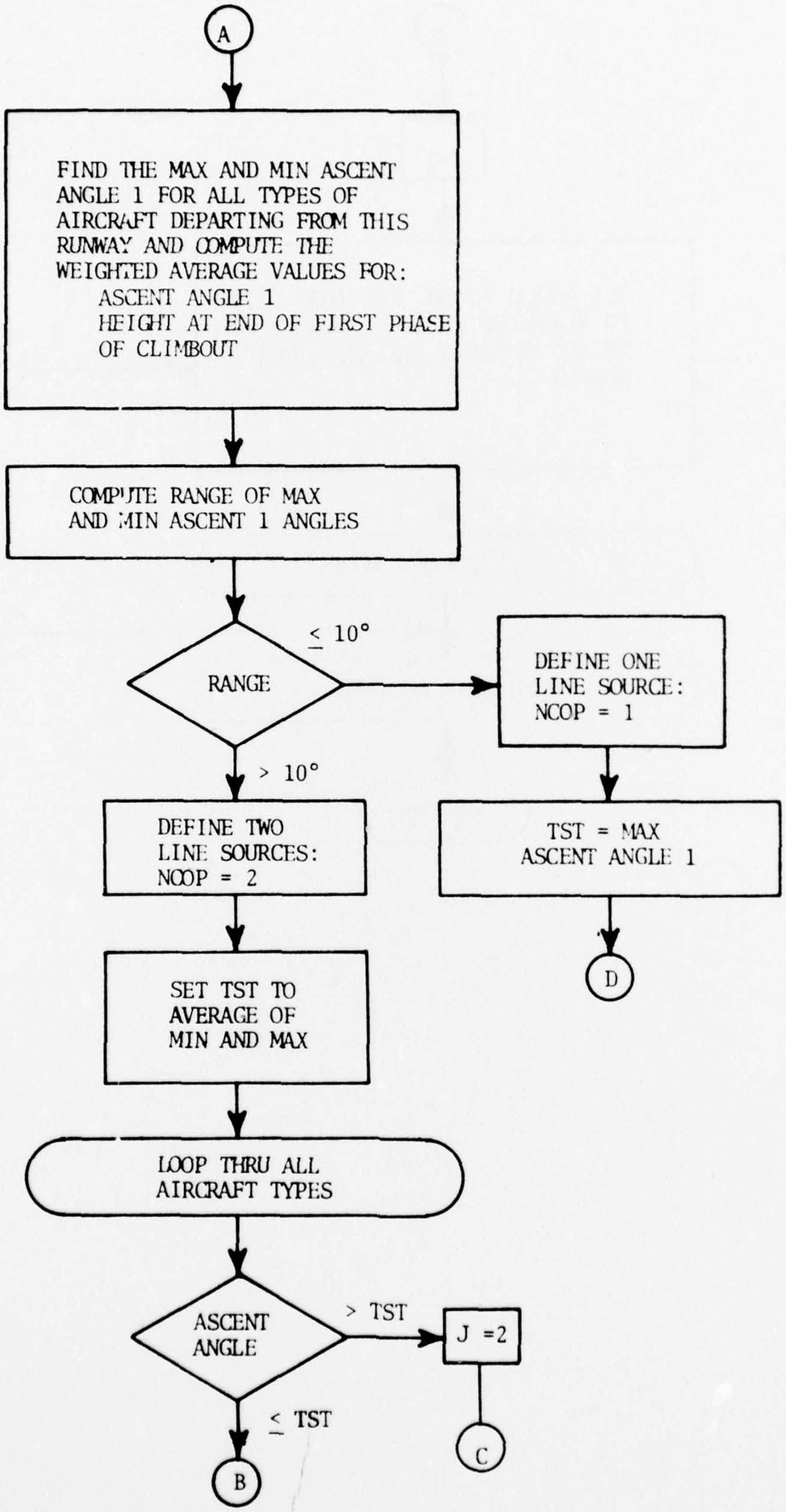
Weighted average values of aircraft activity on the runways.

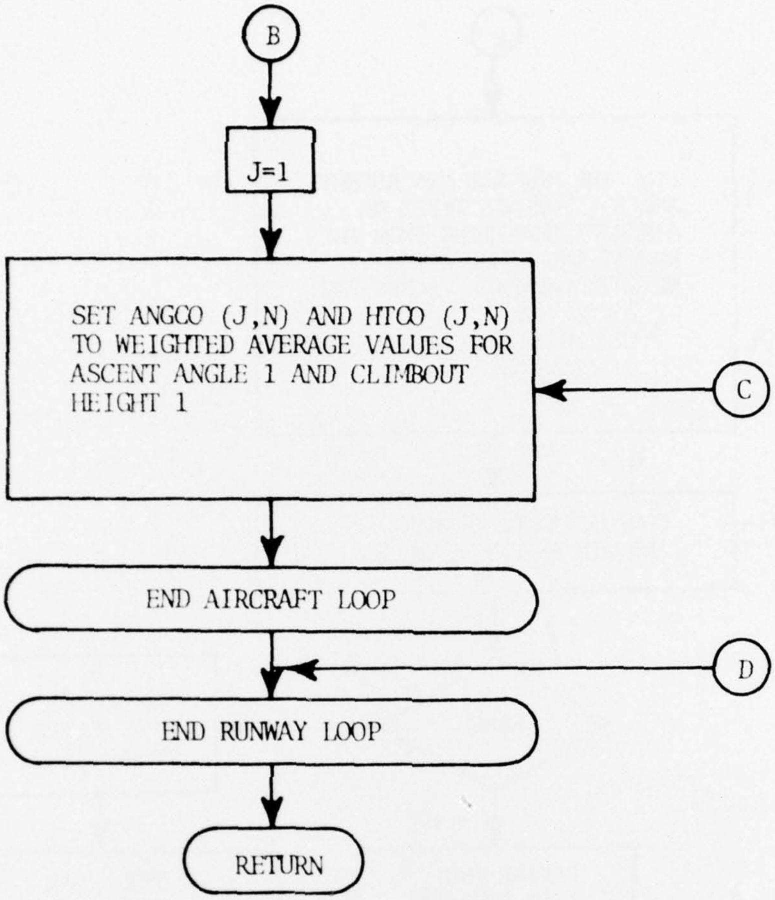
Subroutines Called:

none

SUBROUTINE FLTPTH


START
INITIALIZE ARRAYS
COMPUTE TOTAL NUMBER OF TRAINING FLIGHTS, ARRIVALS AND DEPARTURES ON EACH RUNWAY
BEGIN LOOP OVER N RUNWAYS
FOR ALL TYPES OF AIRCRAFT ARRIVING ON THIS RUNWAY, COMPUTE THE WEIGHTED AVERAGE VALUES FOR:
LANDING VELOCITY
TAXI VELOCITY
ALTITUDE AT START OF FINAL APPROACH
DESCENT ANGLE 2
ADDITIONALLY FOR TRAINING FLIGHTS:
ASCENT ANGLE 1
HEIGHT AT END OF FIRST
PHASE OF CLIMBOUT
INITIALIZE VALUES
A


A
FIND THE MAX AND MIN ASCENT ANGLE 1 FOR ALL TYPES OF AIRCRAFT DEPARTING FROM THIS RUNWAY AND COMPUTE THE WEIGHTED AVERAGE VALUES FOR:
ASCENT ANGLE 1
HEIGHT AT END OF FIRST PHASE OF CLIMBOUT
COMPUTE RANGE OF MAX AND MIN ASCENT 1 ANGLES
RANGE
≤ 10°
DEFINE ONE LINE SOURCE: NCOP = 1
TST = MAX ASCENT ANGLE 1
D
> 10°
DEFINE TWO LINE SOURCES: NCOP = 2
SET TST TO AVERAGE OF MIN AND MAX
LOOP THRU ALL AIRCRAFT TYPES
ASCENT ANGLE
> TST
J =2
C
≤ TST
B


B
J=1
SET ANGCO (J,N) AND HTCO (J,N) TO WEIGHTED AVERAGE VALUES FOR ASCENT ANGLE 1 AND CLIMBOUT HEIGHT 1
C
END AIRCRAFT LOOP
D
END RUNWAY LOOP
RETURN

```
      SUBROUTINE FLTPTH                                                 FTPTH000
C                                                                       FTPTH001
C   THIS ROUTINE DEFINES THE FLIGHT PATH PARAMETERS FOR                 FTPTH002
C   THE APPLICATIONS MODEL                                              FTPTH003
C                                                                       FTPTH004
      REAL INDSPD                                                       FTPTH005
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),    FTPTH006
     . INDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),      FTPTH007
     . COSPD2(8),SRTUPA(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),    FTPTH008
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)                       FTPTH009
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),  FTPTH010
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),   FTPTH011
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),  FTPTH012
     . ACSPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),     FTPTH013
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),  FTPTH014
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),    FTPTH015
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)FTPTH016
      COMMON /MODSIM/       NRRL,NCOP,TST,DISA,RDP(6),RAR(6),RTG(6),   FTPTH017
     1 HTAPP(6),HTTGO(6),VELIND(6),VELTXI(6),ANGAPP(6),ANGTGO(6),      FTPTH018
     2 DIST(2,6),TOPT(2,6),RRTO(2,6),RRTXI(2,6),HTCO(2,6),ANGCO(2,6),  FTPTH019
     3 FRAC(8)                                                          FTPTH020
      DIMENSION WT(2)                                                   FTPTH021
C                                                                       FTPTH022
C   INITIALIZE ARRAYS                                                   FTPTH023
C                                                                       FTPTH024
      DO 5 N=1,NRNWYS                                                   FTPTH025
      RDP(N)=0.0                                                        FTPTH026
      RAR(N)=0.0                                                        FTPTH027
      RTG(N)=0.0                                                        FTPTH028
      ANGAPP(N)=0.0                                                     FTPTH029
      ANGTGO(N)=0.0                                                     FTPTH030
      HTAPP(N)=0.0                                                      FTPTH031
      HTTGO(N)=0.0                                                      FTPTH032
      VELIND(N)=0.0                                                     FTPTH033
      VELTXI(N)=0.0                                                     FTPTH034
C                                                                       FTPTH035
C   COMPUTE TOTAL NUMBER OF TRAINING FLIGHTS, ARRIVALS AND             FTPTH036
C   DEPARTURES ON EACH RUNWAY                                           FTPTH037
C                                                                       FTPTH038
      DO 5 I=1,NACTYP                                                   FTPTH039
      IF(FRAC(I).LE.0.0) GO TO  5                                       FTPTH040
      RTG(N)=RTG(N)+ARRFCN(23,I,N)*ANNTGO(I)                            FTPTH041
      RAR(N)=RAR(N)+ARRFCN(23,I,N)*ANNARR(I)                            FTPTH042
      RDP(N)=RDP(N)+DEPFCN(23,I,N)*ANNDEP(I)                            FTPTH043
    5 CONTINUE                                                          FTPTH044
C                                                                       FTPTH045
C   BEGIN RUNWAY LOOP                                                   FTPTH046
C                                                                       FTPTH047
      DO 10 N=1,NRNWYS                                                  FTPTH048
      DO 11 I=1,NACTYP                                                  FTPTH049
      IF(ARRFCN(23,I,N).LE.0.0)  GO TO 11                               FTPTH050
      IF(FRAC(I).LE.0.0)  GO TO 11                                      FTPTH051
      ARR=0.0                                                           FTPTH052
      ATG=0.0                                                           FTPTH053
      ACTT=0.0                                                          FTPTH054
      IF(RAR(N).LE.0.0)  GO TO 6                                        FTPTH055
      ARR=ARRFCN(23,I,N)*ANNARR(I)/RAR(N)                               FTPTH056
      ACTT=ACTT+1.0                                                     FTPTH057
    6 IF(RTG(N).LE.0.0)  GO TO 7                                        FTPTH058
      ATG=ARRFCN(23,I,N)*ANNTGO(I)/RTG(N)                               FTPTH059
      ACTT=ACTT+1.0                                                     FTPTH060
    7 ACT=(ARR+ATG)/ACTT                                                FTPTH061
```

```
      IF (ACT.LE.0.0) GO TO 11
C
C   COMPUTE WEIGHTED AVERAGE VALUES FOR ALL AIRCRAFT
C   ARRIVING ON THIS RUNWAY
C
      VELTXI(N)=VELTXI(N)+ACT*TXISPD(I)
      VELLND(N)=VELLND(N)+ACT*LNDSPD(I)
      HTAPP(N)=HTAPP(N)+ACT*APPHT2(I)
      ANGAPP(N)=ANGAPP(N)+ACT*DSCNT2(I)
      ANGTGC(N)=ANGTGC(N)+ATG*ASCNT1(I)
      HTTGO(N)=HTTGO(N)+ATG*COHT1(I)
   11 CONTINUE
C
C   INITIALIZE ARRAYS
C
      DO 12 J=1,2
      WT(J)=0.0
      HTCC(J,N)=0.0
   12 ANGCC(J,N)=0.0
      HGT=0.0
      AAVE=0.0
      PHIMAX=0.0
      PHIMIN=175.
C
C   FIND MAX AND MIN ASCENT ANGLE 1
C
      DO 13 I=1,NACTYP
      IF(RDP(N).LE.0.0) GO TO 10
      IF(FRAC(I).LE.0.0) GO TO 13
      DEP=DEPFCN(23,I,N)*ANNDEP(I)/RDP(N)
      IF (DEP.LE.0.0) GO TO 13
      IF (ASCNT1(I).GT.PHIMAX) PHIMAX=ASCNT1(I)
      IF (ASCNT1(I).LT.PHIMIN) PHIMIN=ASCNT1(I)
C
C   COMPUTE WEIGHTED AVERAGE VALUES FOR ALL AIRCRAFT
C   DEPARTING FROM THIS RUNWAY
C
      AAVE=DEP*ASCNT1(I)+AAVE
      HGT=HGT+DEP*COHT1(I)
   13 CONTINUE
C
C   COMPUTE RANGE OF MAX AND MIN CLIMBOUT PHASE 1
C
      RANGE=PHIMAX-PHIMIN
      NCOP=1
      IF (RANGE.GT.0.174533) GO TO 14
C
C   RANGE IS LESS THAN 10 DEGREES.  DEFINE ONE LINE SOURCE
C   FOR ALL AIRCRAFT
C
      ANGCC(1,N)=AAVE
      HTCC(1,N)=HGT
      TST=PHIMAX
      GO TO 16
C
C   RANGE IS GREATER THAN 10 DEGREES.  DEFINE TWO LINE SOURCES
C   DEPENDING ON AIRCRAFT HAVING A CLIMBOUT PHASE 1 ANGLE LYING IN
C   THE LOWER OR UPPER HALF OF THE RANGE FOR THIS RUNWAY
C
   14 NCOP=2
      TST=PHIMIN+RANGE/2.
      DO 15 I=1,NACTYP
      IF(FRAC(I).LE.0.0) GO TO 15                                       FTPTH124
      DEP=DEPFCN(23,I,N)*ANNDEP(I)/RDP(N)                               FTPTH125
      IF (DEP.LE.0.0) GO TO 15                                          FTPTH126
      J=1                                                               FTPTH127
      IF (ASCNT1(I).GT.TST) J=2                                         FTPTH128
      WT(J)=WT(J)+DEP                                                   FTPTH129
      ANGCO(J,N)=ANGCO(J,N)+DEP*ASCNT1(I)                               FTPTH130
      HTCO(J,N)=HTCO(J,N)+DEP*COHT1(I)                                  FTPTH131
   15 CONTINUE                                                          FTPTH132
      DO 17 J=1,2                                                       FTPTH133
      HTCO(J,N)=HTCO(J,N)/WT(J)                                         FTPTH134
   17 ANGCO(J,N)=ANGCO(J,N)/WT(J)                                       FTPTH135
   16 CONTINUE                                                          FTPTH136
   10 CONTINUE                                                          FTPTH137
      RETURN                                                            FTPTH138
      END                                                               FTPTH139
```

SUBROUTINE TODIST

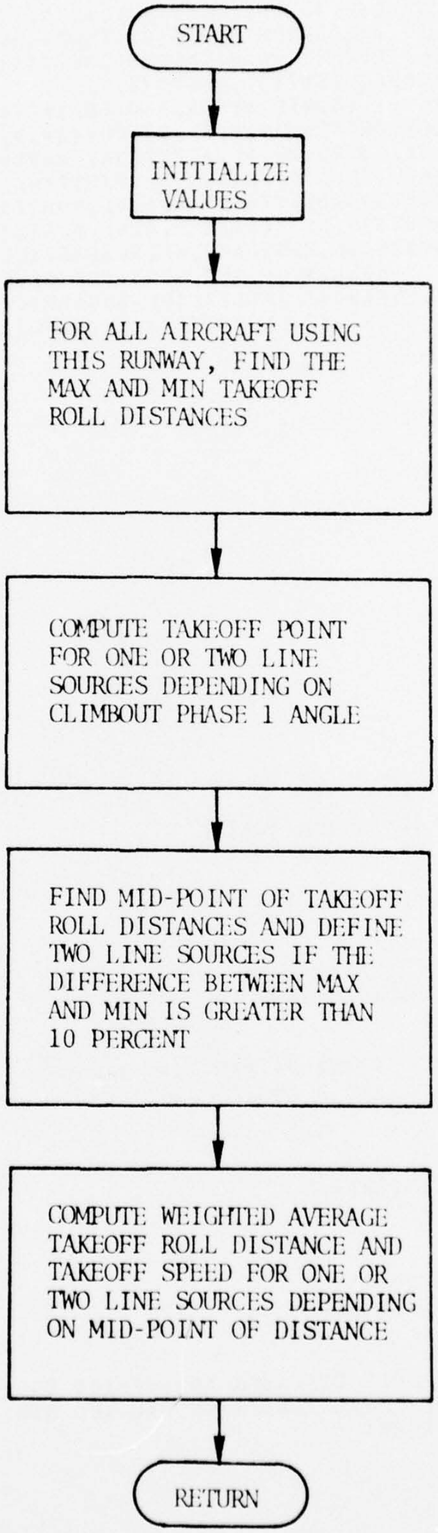
Purpose: To define the weighted average takeoff roll distances and speeds for a given runway.

Input: Aircraft activity parameters on the runway.

Output: Weighted average takeoff roll distances and speeds.

Subroutines Called: None

SUBROUTINE TODIST(N)


START
INITIALIZE VALUES
FOR ALL AIRCRAFT USING THIS RUNWAY, FIND THE MAX AND MIN TAKEOFF ROLL DISTANCES
COMPUTE TAKEOFF POINT FOR ONE OR TWO LINE SOURCES DEPENDING ON CLIMBOUT PHASE 1 ANGLE
FIND MID-POINT OF TAKEOFF ROLL DISTANCES AND DEFINE TWO LINE SOURCES IF THE DIFFERENCE BETWEEN MAX AND MIN IS GREATER THAN 10 PERCENT
COMPUTE WEIGHTED AVERAGE TAKEOFF ROLL DISTANCE AND TAKEOFF SPEED FOR ONE OR TWO LINE SOURCES DEPENDING ON MID-POINT OF DISTANCE
RETURN

```
      SUBROUTINE TODIST(N)                                              TDIST000
C     ROUTINE TO SETUP TAKEOFF ROLL DISTANCES FOR APPLICATIONS MODEL    TDIST001
      COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),     TDIST002
     . INDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),       TDIST003
     . COSPD2(8),SHTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),     TDIST004
     . APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2)                        TDIST005
      COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),   TDIST006
     . ANNDEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),    TDIST007
     . DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),   TDIST008
     . ACSEIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),      TDIST009
     . IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),   TDIST010
     . IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),     TDIST011
     . PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8) TDIST012
      COMMON /MODSIM/       NRRL,NCOP,TST,DISA,RDP(6),RAR(6),RTG(6),    TDIST013
     . HTAPP(6),HITGO(6),VELLND(6),VELTXI(6),ANGAPP(6),ANGTGO(6),       TDIST014
     . DIST(2,6),TOPT(2,6),RRTO(2,6),RRTXI(2,6),HTCO(2,6),ANGCO(2,6),   TDIST015
     . FRAC(8)                                                          TDIST016
      DIMENSION WT(2)                                                   TDIST017
C                                                                       TDIST018
C    INITIALIZE VALUES                                                  TDIST019
C                                                                       TDIST020
      DISA=0.0                                                          TDIST021
      DISMAX=0.                                                         TDIST022
      DISMIN=1.E10                                                      TDIST023
      DO 5 J=1,2                                                        TDIST024
      WT(J)=0.0                                                         TDIST025
      DIST(J,N)=0.0                                                     TDIST026
      TOPT(J,N)=0.0                                                     TDIST027
      RRTO(J,N)=0.0                                                     TDIST028
    5 RRTXI(J,N)=0.0                                                    TDIST029
C                                                                       TDIST030
C    BEGIN AIRCRAFT LOOP                                                TDIST031
C                                                                       TDIST032
      DO 10 I=1,NACTYP                                                  TDIST033
      IF(DEPFCN(23,I,N).LE.0.0) GO TO 10                                TDIST034
      IF(FRAC(I).LE.0.0) GO TO 10                                       TDIST035
      DEP=DEPFCN(23,I,N)*ANNDEP(I)/RDP(N)                               TDIST036
      IF (DEP.LE.0.0) GO TO 10                                          TDIST037
C                                                                       TDIST038
C    FIND MAX AND MIN TAKEOFF ROLL DISTANCES                            TDIST039
C                                                                       TDIST040
      RRDS=DEPFCN(5,I,N)                                                TDIST041
      IF (RRDS.GT.DISMAX) DISMAX=RRDS                                   TDIST042
      IF (RRDS.LT.DISMIN) DISMIN=RRDS                                   TDIST043
      DISA=DISA+RRDS*DEP                                                TDIST044
C                                                                       TDIST045
C    COMPUTE TAKEOFF POINT FOR ONE OR TWO LINE SOURCES                  TDIST046
C    DEPENDING ON CLIMBOUT PHASE 1 ANGLE                                TDIST047
C                                                                       TDIST048
      J=1                                                               TDIST049
      IF (ASCNT1(I).GT.TST) J=2                                         TDIST050
      TOPT(J,N)=TOPT(J,N)+DEP*RRDS                                      TDIST051
      WT(J)=WT(J)+DEP                                                   TDIST052
   10 CONTINUE                                                          TDIST053
      DO 11 J=1,2                                                       TDIST054
      IF(WT(J).LE.0.0) GO TO 11                                         TDIST055
      TOPT (J,N)=TOPT(J,N)/WT(J)                                        TDIST056
   11 WT(J)=0.0                                                         TDIST057
C                                                                       TDIST058
C    FIND MIDPOINT TAKEOFF ROLL DISTANCE AND DEFINE TWO                 TDIST059
C    LINE SOURCES IF THE DIFFERENCE BETWEEN MAX AND MIN                 TDIST060
C    IS GREATER THAN 10 PERCENT                                         TDIST061
```

```
C
      DBAR=(DISMAX-DISMIN)/2.
      NRRL=1
      IF (DISMIN.LT.0.9*DISMAX) NRRL=2
C
C    BEGIN AIRCRAFT LOOP
C
      DO 15 I=1,NACTYP
      IF(DEPFCN(23,I,N).LE.0.0) GO TO 15
      IF(FRAC(I).LE.0.0) GO TO 15
      DEP=DEPFCN(23,I,N)*ANNDEP(I)/RDP(N)
      IF (DEP.LE.0.0) GO TO 15
      RRDS=DEPFCN(5,I,N)
      DEL=RRDS-DBAR
      J=1
      IF (DEL.GT.0.0.AND.DBAR.GT.0.0) J=2
      WT(J)=WT(J)+DEP
      DIST(J,N)=DIST(J,N)+DEP*RRDS
      RRTO(J,N)=RRTO(J,N)+DEP*TOSPD(I)
   15 CONTINUE
C
C    COMPUTE WEIGHTED AVERAGE TAKEOFF ROLL DISTANCE AND
C    TAKEOFF SPEED FOR NRRL LINE SOURCES
C
      DO 17 J=1,NRRL
      DIST(J,N)=DIST(J,N) / WT(J)
      RRTO(J,N)=RRTO(J,N)/ WT(J)
   17 CONTINUE
      RETURN
      END
```

REFERENCES

1. Bingaman, D. J., and L. E. Wangen, "Air Quality Assessment Model for Air Force Operations - Source Emissions Inventory Computer Code Documentation," Air Force Civil and Environmental Engineering Development Office report number CEEDO-TR-76-33, April 1977.

2. Bingaman, D. J., "Air Quality Assessment Model for Air Force Operations - Short Term Emission/Dispersion Computer Code Documentation," Air Force Civil and Environmental Engineering Development Office report number CEEDO-TR-76-34, April 1977.

3. Menicucci, D. F., "Air Quality Assessment Model (AQAM). Data Reduction and Operations Guide," Air Force Weapons Laboratory report number AFWL-75-307, Oct 1976.

4. Rote, D. M., and L. E. Wangen, "A Generalized Air Quality Assessment Model for Air Force Operations - Technical Report," Air Force Weapons Laboratory Report No. AFWL-TR-74-304, February 1975.

INITIAL DISTRIBUTION

Hq USAF/PREVP	3
Hq TAC/DEEV	2
CINCUSAFE/Surgeon	1
USAFA/DEV	1
AFIT/DEM	1
OEHL/CC	1
OEHL/OL-AA	5
ADTC/CSV	1
DDC/TCA	12
EA/USA Environ Hygn Agency	1
USA CERL	1
WESTDIV/Code 09BE	1
Technology Transfer Staff/EPA	1
Office of R&D/EPA	1
NARF/Code 64270	2
Nav Postgrad School/Code 41A	2
Argonne Nat Lab/EESD	5
US Environ Protection Agency	2
FAA/AEQ	1
Dr Delaney	1
Det 1 Hq ADTC/WE	2
Det 1 Hq ADTC/EC	1
Det 1 Hq ADTC/ECA	7
AFCEC/DEE	3
Det 1 Hq ADTC/PRT	1
AFATL/DLODR	1

(The reverse of this page is blank)